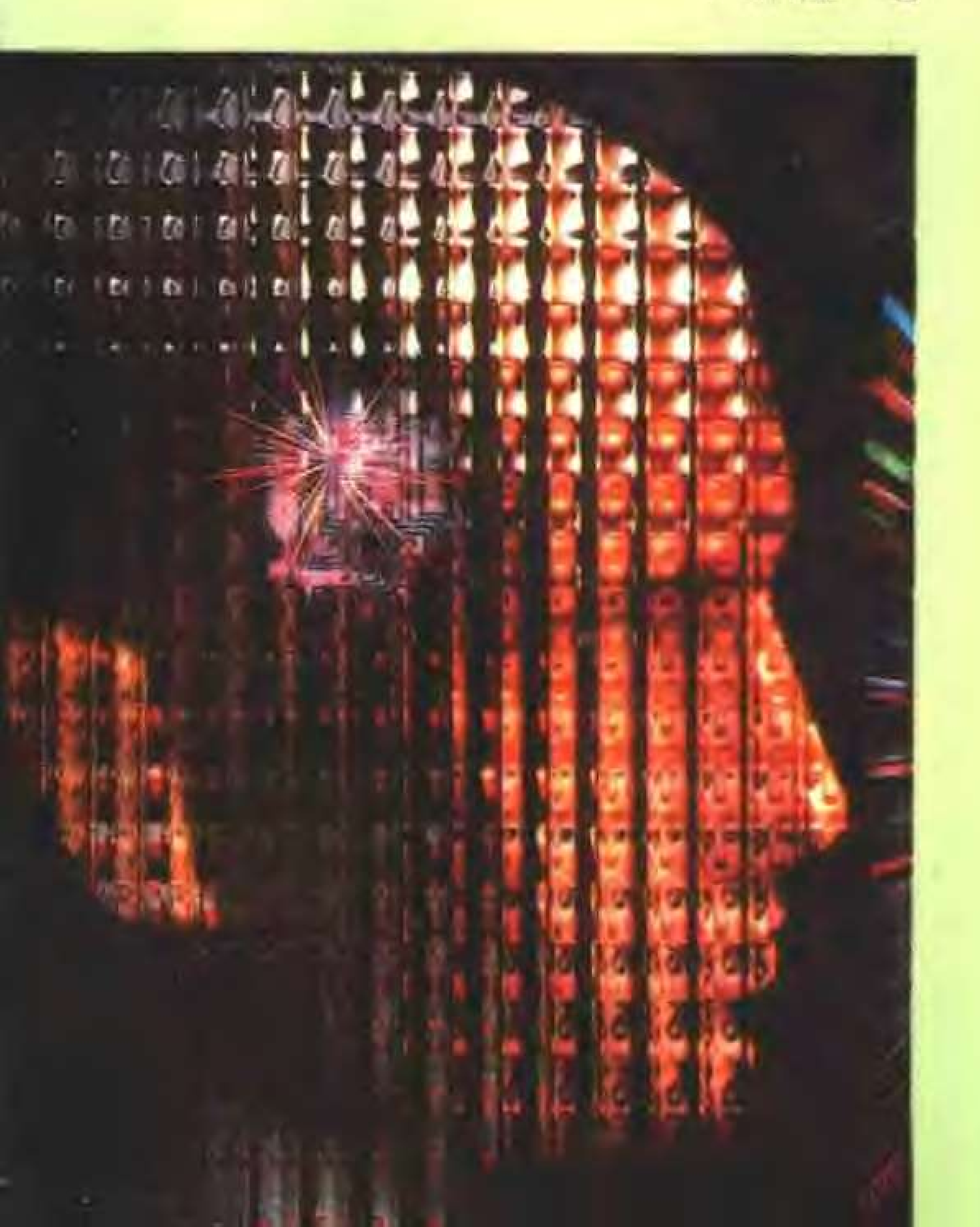
计算密码学

COMPUTATIONAL CRYPTOGRAPHY

走向数学丛书

卢开澄 著





责任编辑：孟实华　装帧设计：邱湘军

内容提要

保密通信是一个十分古老而又引人入胜的课题，在国防、商业、通信以及经济生活各方面有重要的意义。随着通信技术的不断发展，特别是计算机的广泛采用，密码的设计思想不断更新，密码体制在不断改进。

本书深入浅出地介绍了密码学的基本知识以及密码学的近代发展。作者不仅介绍了一些古典的加密方法，更着重讲述了70年代以来发展的 DES 数据加密标准和公开密钥体制，包括公钥体制的各种方案和与此有关的新的研[illegible]课题，如大数分解、数字签名、离散对数、[illegible]知识证明，纠错码加密方法等，并有许多例[illegible]供读者参考。书中还论述了数学的许多分[illegible]（如概率统计、信息论、数论、置换群和有限[illegible]理论、组合学以及算法复杂性理论等）的思想[illegible]方法和结果（包括许多近代的理论数学成果[illegible]的相互交织，以及在保密通信领域中的应用。

ISBN7—5355—1584—3/G・1579

湘教（92）30期　定价：3.90元

（湘）新登字005号

走向数学丛书

计算密码学

卢开澄 著

湖南教育出版社

前　言

王　元

从力学、物理学、天文学直到化学、生物学、经济学与工程技术，无不用到数学. 一个人从入小学到大学毕业的十六年中，有十三、四年有数学课. 可见数学之重要与其应用之广泛.

但提起数学，不少人仍觉得头痛，难以入门，甚至望而生畏. 我以为要克服这个鸿沟，还是有可能的. 近代数学难于接触，原因之一大概是由于其符号、语言与概念陌生，兼之近代数学的高度抽象与概括，难于了解与掌握. 我想，如果知道讨论的对象的具体背景，则有可能掌握其实质. 显然，一个非数学专业出身的人，要把数学专业的教科书都自修一遍，这在时间与精力上都不易做到. 若停留在初等数学水平上，哪怕做了很多难题，似亦不会有助于对近代数学的了解. 这就促使我们设想出一套“走向数学”小丛书，其中每本小册子尽量

用深入浅出的语言来讲述数学的某一问题或方面，使工程技术人员，非数学专业的大学生，甚至具有中学数学水平的人，亦能懂得书中全部或部分含义与内容. 这对提高我国人民的数学修养与水平，可能会起些作用. 显然要将一门数学深入浅出地讲出来，决非易事. 首先要对这门数学有深入的研究与透彻的了解. 从整体上说，我国的数学水平还不高，能否较好地完成这一任务还难说. 但我了解很多数学家的积极性很高，他们愿意为“走向数学”撰稿. 这很值得高兴与欢迎.

承蒙国家自然科学基金委员会、中国数学会数学传播委员会与湖南教育出版社支持，得以出版这套“走向数学”丛书，谨致以感谢.

前　言

在信息时代的今天，信息和其它资源一样成为人类重要的财富，而且信息本身还是时间，甚至于是生命．当前计算机被广泛应用而且日渐深入，几乎已无处不感到它的存在．但储存信息的计算机系统是很脆弱的，信息的传输则是依靠“不设防”的公共信道．信息作为一种资源被盗窃不同于其它的财富，首先不易被发现，而其后果可能更为严重．如何保护信息的安全？这个问题便尖锐地被提到议事日程上来．对它进行加密看来是有效而且可行的一种方法．它只需要付出很少的代价，便可得到满意的结果．

密码作为军事和政治斗争中的一种技术，已有悠久的历史，而且也有其不寻常的表现．但成为一门学科，还只是近几年的事．在计算机科学蓬勃发展的刺激下，70年代中期在这块园地开出两朵艳丽夺目的花，一是公钥密码系统，一个是数据加密标准DES．从此奠定了近代密码学的基础．在短短十多年时间里其发展可以说是异彩纷呈，令人目不暇接．数学在此过程中扮演了一个重要的角色．我们都听说数论中“1+1”问题如何

如何，但数论在近代密码学中的贡献更令人神往．读者将看到除数论外，信息论，概率统计，抽象代数中的群论和有限域理论，组合论，算法复杂性理论，编码理论，自动机理论，甚至于代数几何中的椭圆曲线等，都在密码学研究中找到了自己的位置．这在其它学科中是不多见的现象．这些内容饶有趣味，本书用通俗的方式讨论它们．密码学的研究成果已不局限于通信保密上，如数字签名，身份验证等技术已是其它学科的基础．

本书分两个部分，密码学的基本概念和近代密码学的若干问题，有一些是我们工作的成果．这样的划分其实是模糊的，比如数据加密标准无疑是近代密码学的一个方面，但放在了第一部分中，只是因为它的基本方法更接近于传统密码．第二部分也不能简单地用公钥来概括．近代密码学的突出特点是更多地依靠计算，所以本书命名为“计算密码学”．作者认为传统密码和公钥密码是密码学中不可截然分开更不是对立的两个部分．

密码学作为一门学科非常年轻，但大有可为，这是现实的需要．我国必须要有自己的数据加密标准，我们不能没有自己的密码系统．作者希望本书对普及和推广这方面的知识会起到抛砖引玉的作用．为了让更多的读者能接受讨论的内容，作者力求做到深入浅出．但内容毕竟涉及到近代密码学的诸多方面，有相当的深度和难度，请读者能理解．

卢开澄

目　录

第一章　密码学若干基本概念

§1 引　论

密码作为一种技术已有上千年的历史. 自从人类有了战争，便自然产生了密码. 然而密码正式成为一门学科，还是近几年的事. 在计算机发展到网络的信息时代，信息本身就是时间，就是财富. 但信息存储于计算机系统中，信息的传输依赖于十分脆弱的公共信道. 信息的泄露不易被发现，但它造成的社会影响是巨大的. 所以，保护信息的安全是时代发展的必然要求，它已被迫切地提到日程上来.

密码是保护信息安全的有效而可行的方法. 它用很小的代价，为信息提供足够的安全保护. 在计算机蓬勃发展的刺激下，数据安全作为一个新的分支已活跃在计算机科学这个领域里. 不仅如此，它还是其他许多学科的基础和工具，被广泛地应用着.

数据加密标准DES和公钥密码体制，是70年代后半期在密码学园地上盛开的两朵奇葩，它们几乎是在相同的时间里提出的. DES是Data Encryption Standard的缩写，由IBM公司研究并提出，1977年经美国国家标准局批准，作为非机密机构保密通信用，最初预定服务期限十年. 至今，十年已经过去了，DES还在“超期服役”中. 1976年狄菲（W. Diffie）和赫尔曼（M. E. Hellman）在一篇著名的论文“New Directions in Cryptagraphy”中提出了公钥密码体制的构想，不久便推出公钥密码系统，可以毫不夸张地说：没有公钥密码和DES的研究，便没有近代密码学. 近代密码学的突出特点是更多地依靠于计算，公钥密码的研究异彩纷呈，当Diffie和Hellman提出他们十分卓越的思想时，还没有一个具体的实例. 但由于它的优势十分明显，所以，在这以后，各种公钥体制纷至沓来，真有点咄咄逼人的气势. 十五年过去了，应该说公钥尚未成功，还要继续努力. 这不能苛求于公钥本身，它毕竟才只有十几年. 十几岁对于一个人来说最多才是青年时期. 虽然如此，公钥密码的研究仍然光彩夺目. 除了计算机科学外，它还涉及数学中的数论、群论、有限域理论、信息论、编码理论、自动机理论、算法复杂性理论、概率统计，以及代数几何中的椭圆曲线等，这在其它学科中也是罕见的现象. 以上各方面在本书中都将一一论及.

讨论近代密码学无疑是本书的重点，公钥自然是中心内容，公钥是相对于传统的密码体制而存在的，所以在第一章里将简单地介绍一些传统的密码技术. 尽管如此，它的方法和引出的数学问题也是饶有趣味的.

我国必须要有自己的密码系统，也要有自己的数据加密标准，这是时代的需要. 而且唯此不能依靠进口，这是学科的性质所决定的. 开展密码的研究是当务之急，它除了依靠专业人

员外，群众性的研究也很重要，国外的经验也说明了这一点.

§2　保密通信是怎样进行的

若 A 要通过公共信道向 B 送去信息 m，由于公共信道缺乏足够的安全保护，信息 m 容易被第三者所窃取，甚至于被篡改，为此在 m 进入公共信道之前，先对它进行加密变换，得密文 c，即：

$$E: c = E_k(m)$$

其中 k 是参数，称为密钥，A 将密文 c 送给 B，B 收到后对 c 作解密变换，恢复 m，即：

$$D: m = D_k(c)$$

所以，加密变换 E 和解密变换 D 实际上是一对变换和它的逆，即：

$$D_k(E_k(m)) = m$$

相对于密文 c，m 称为明文.

加密通信的过程可用下图来表示：

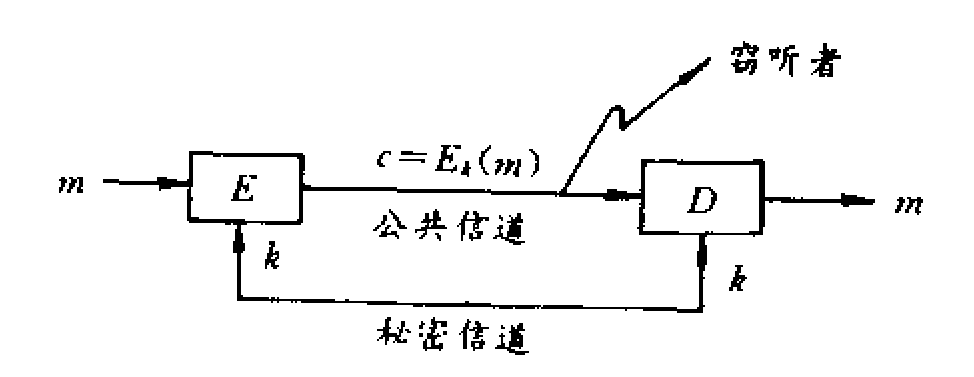


图 1.2.1

密钥 k 在保密通信中占有极其重要的地位. 它由通信双方秘密商定，只有他们双方掌握，第三者就是知道所用的加密算法 E 和解密算法 D，若不知道所用的密钥 k，也仍然无法获得明文 m，以此来达到保密的目的. 保密只是通信安全的一个目的. 还有一个目的是信息 m 的完整性，即要保证 B 收到的密文 c 不会被第三者篡改，即第三者若不掌握密钥 k，就无法伪造任何密文. 下面举例说明以上的概念.

例 1 凯撒 (Caesar) 密码 凯撒密码是将明文的每一个字母一律循环推移 k 位. 所以，凯撒密码也叫做单表密码，例如明文：

Secure message transmission is of extreme importance in information based society.

这段明文的意思是：在信息社会里，秘密通信是极其重要的.

现将明文字符通过凯撒密码后移 3 位加密得密文：

VHFXUH PHVVDJH WUDQVPLVVLRQ LV RI HAWUHPH LPSRUWDQFH LQ LQIRUPDWLRQ EDVHG VRFLHWB

这里凯撒变换是加密算法，k 是密钥. 本例 $k=3$. 可见凯撒密码并不安全，也就是说截到密文 c，在不知密钥的前提下也不难获得明文 m.

例 2 词组密钥密码 明文字母和密文的置换如下表：

a b c d e f g h i j k l m n o p q r s t u v w x y z
F I V E S T A R B C D G H J K L M N O P Q U W X Y Z

即字母表 *abcde……uvwxyz* 分别和 *FIVESTAR……YZ* 对应，密钥为词组 *five stars*，这样明文：

Secure message transmission is of extreme importance in

information based society.

被加密成密文：

OSVQNS HSOOFAS PNFJOHBOOBKJ BO KT SXPN-SHS BHLKNPFJVS BJ BJTKNHFPBKJ IFOSE OKVBSPY.

要对词组密钥密码系统的密钥采取强行(brute force)搜索的方法，则要面临26个字母的全排列的各种可能，根据Stirling公式：

$$n! = \sqrt{2n\pi}\left(\frac{n}{e}\right)^n$$

可求得　$26! \approx 4 \times 10^{26}$

若以每秒可搜索 10^7 种排列的超高速电子计算机进行穷举，需要的时间为

$$\begin{aligned} T &= 4 \times 10^{26}/(365 \times 24 \times 3600 \times 10^6) \\ &= 4 \times 10^{26}/(3.1536 \times 10^{12}) \\ &= 1.27 \times 10^{13}(\text{年}) \end{aligned}$$

显然，采用蛮攻击或强行搜索的方法是行不通的．在下一节里将介绍一种统计分析的方法．

例3　蒲内费厄(Playfair)密码

蒲内费厄是英国的密码学家．图1.2.2是个5×5方阵．

T	S	I	N	G
H	U	A	V	E
R	Y	B	C	D
F	K	L	M	O
P	Q	W	X	Z

图1.2.2

矩阵中元素TSINGHUAVERYBCDFKLMOPQWXZ正好包含除了J以外的其它25个英文字母．前面12个字母是依TS-INGHUA UNIVERSITY顺序取出前面未出现的字母，后面跟

以依英文字母表中前面未出现的英文字母. 加密算法由明文字母对 m_1m_2 而定. 设 m_1m_2 的密文为 c_1c_2. 加密法则如下:

(a) 若 m_1 和 m_2 在同一行,则 c_1 和 c_2 分别是 m_1 和 m_2 右边的字母, 这里将第 1 列看作是在第 4 列的右边.

(b) 若 m_1 和 m_2 在同一列,则 c_1 和 c_2 分别为 m_1 和 m_2 下方的字母. 第 1 行看作是位于第 4 行的下方.

(c) 若 m_1 和 m_2 不在同一行或同一列,则 c_1 和 c_2 为矩形的两个顶点,且该矩形的其它两顶点为 m_1 和 m_2,且 c_1 和 m_1 同行, c_2 和 m_2 同行.

(d) 若 $m_1=m_2$, 则在明文 m_1 和 m_2 之间插入一空字符(设为 X).

(e) 若明文字符数是奇数,则在明文末端加上一空字符.

例 4 利用图 1.2.2 的方阵, 对明文

BEIJING CHINA

进行加密. 在这里 J 当作 I 处理,与图 1.2.2 对应的密钥为 *TSINGHUA UNIVERSITY* 先将明文分成两个字符一组

BE IX IX IN GC HI NA

分别加密得

DA NW NW NG ND AT IV

瀟内费厄密码加密的结果, 一个字母对应的密文并不固定, 这不同于词组密钥密码系统.

例 5 费尔南(Vernam)密码

费尔南密码假定明文 m 用 n 位的 0,1 符号串来表示,密钥 k 也是 0, 1 符号串, 设

$$m = m_1m_2\cdots m_n,\ m_i = 0 \text{ 或 } 1,\ i = 1,2,\cdots,n$$

$$k = k_1k_2,\cdots k_l,\ k_j = 0 \text{ 或 } 1,\ j = 1,2,\cdots,l$$

$$E_k(m) = c = c_1c_2\cdots c_n,\ c_i \equiv m_i + k_i \pmod 2$$

$$i = 1,2,\cdots,n$$

所以，费尔南密码的弱点在于若已知密钥 k 的一组明文和它对应的密文，则费尔南密码便被攻破.

设密文

$$c_l = c_1^{(l)} c_2^{(l)} \cdots c_n^{(l)},\ l = 1,2$$

分别对应于明文

$$m_l = m_1^{(l)} m_2^{(l)} \cdots m_n^{(l)},\ l = 1,2$$

即 $\quad c_i^{(l)} = k_i \oplus m_i^{(l)},\quad l = 1,2,\ i = 1,2,\cdots,n$

则 $\quad c_i^{(1)} \oplus c_i^{(2)} = m_i^{(1)} \oplus k_i \oplus m_i^{(2)} \oplus k_i$

$$= m_i^{(1)} \oplus m_i^{(2)}$$

若 $m_i^{(1)}$ 已知，则 $m_i^{(2)}$ 便可得到.

一般地，密钥 k 的长度 l 有限，可以周而复始重复地出现.也可以用长度为 l_1 和 l_2 的两个密钥

$$k_1 = k_1^{(1)} k_2^{(1)} \cdots k_{l_1}^{(1)}$$

$$k_2 = k_1^{(2)} k_2^{(2)} \cdots k_{l_2}^{(2)}$$

只要 l_1 和 l_2 互素，由 k_1 和 k_2 可以产生周期为 $l_1 l_2$ 的密钥比特流

$$K = k_1 k_2 \cdots k_n,\quad n = l_1 l_2$$

$$k_i = k_i^{(1)} \oplus k_i^{(2)},\quad i = 1,2,\cdots,l_1 l_2$$

这里 $k_i^{(1)}$ 和 $k_i^{(2)}$ 都分别是由 k_1 和 k_2 产生的周期比特流.

若费尔南密码的密钥 k 是一组不重复的随机流，则这样的密码称之为一次一密. 一次一密密码是完全保密密码. 完全保密的概念见第一章§8商农理论.

§3 统计分析法

在密码学中加密和破译是一对矛和盾，所以，了解破译技

术对于研究密码也是必不可少的. 在词组密钥的密码系统中，经过字母的变换使得明文面目全非，然而并非没有留下蛛丝马迹. 例如英文字母出现的频率差别很大，这就给密码分析者以可乘之机. 大量的统计表明，虽然统计的对象迥异，但统计结果表明，各个字母各自出现的频率却惊人地接近.

下面是一组统计结果

a：0.0856	b：0.0139	c：0.0279	d：0.0378
e：0.1304	f：0.0289	g：0.0199	h：0.0528
i：0.0627	j：0.0013	k：0.0042	l：0.0339
m：0.0249	n：0.0707	o：0.0797	p：0.0199
q：0.0012	r：0.0677	s：0.0007	t：0.1045
u：0.0249	v：0.0092	w：0.0149	x：0.0017
y：0.0199	z：0.0008		

图 1.3.1 是它们的频率图：

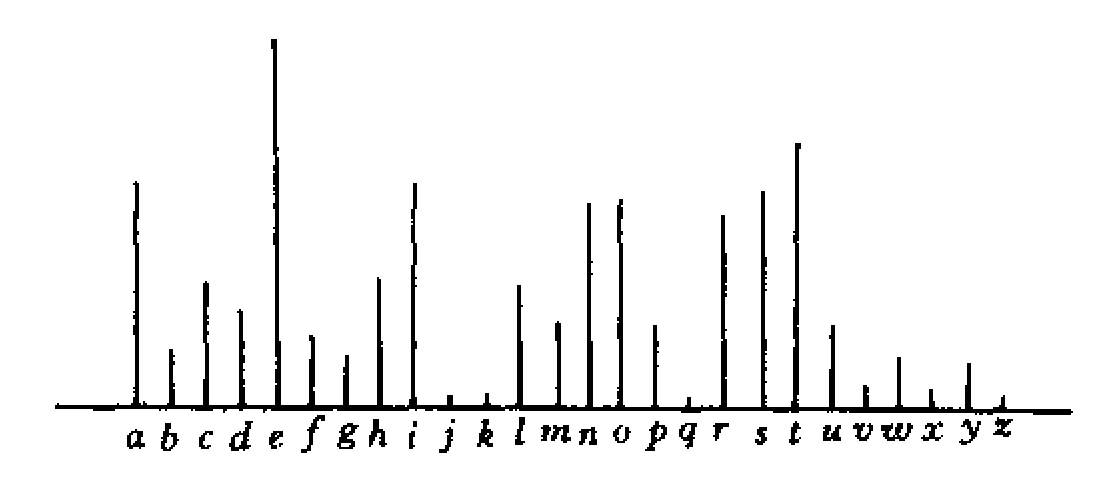


图 1.3.1

从频率图可以看出以下几个比较明显的差异：

1. 字母 e、t，a，o，n，i，r，s，h 的频率较高，其中 e 尤为突出.

2. 字母 d，l，u，c，m 是中频.

3. 字母 p，t，y，w，g，b，v 属于低频.

4. 字母 j，k，q，x，z 为稀少字母.

我们说它们之间有比较明显的差异，以高频字母中频率最低的 h 和中频字母中频率最高的 d 为例，频率差别为：$0.0528-0.0378=0.015$. 英文中许多差异给破译者提供了线索.

下面是一组由词组密钥密码系统加密而得的密文：

EQNBJ APRSG FOPPS JYSFN OPRSF
NPKTO SVQNB JAFGG TKNHO KTEFP
FBJVG QEBJA EBABP FGOLS SVRRF
OBHLN KUSEH FJYTK GEPRS LNBHF
NYNSF OKJTK NPRBO RFOIS SJPRS
FEUSJ PKTHB VNKSG SVPNK JBVOP
RSVKH IGSXB PYKTP RSTQJ VPBKJ
OPRFP VFJJK WISLS NTKNH SEIYP
RSHFV RBJSR FONBO SJENF HFPBV
FGGYF OFNSO QGPKT PRBON SVSJP
ESUSG KLHSJ PBJPS VRJKG KAYHF
JYKTP RSVBL RSNOY OPSHP RFPWS
NSKJV SVKJO BESNS EOSVQ NSFNS
JKWIN SFDFI GS

(1) 对这 327 个字母进行统计得：

A：4	B：20	C：0	D：1
E：11	F：26	G：14	H：12
I：6	J：23	K：22	L：5
M：0	N：24	O：20	P：27
Q：6	R：20	S：43	T：10
U：3	V：17	W：3	X：1

Y：9　　Z：0

(2)　其中 S 的出现次数比其它字母明显的多故可知密文 $S\leftrightarrow e$，即 S 可能和明文字母 e 对应.

(3)　除 S 预计和明文字母 e 对应外，其余前 8 个出现次数最多的字母分别为：

P:27，F:26，N:24，K:23，J:22，B:21，O:20，R:20

初步判断这 8 个字母以某种方式和 t，a，o，n，i，r，s，h 相对应. 它们之间的前后连缀特点可以加以利用.

(4)　统计 S 和 P，F，N，K，J，B，O，R 的前后缀的数目如下：

↓ / →	P	F	N	K	J	B	O	R
S	3	0	7	1	0	0	3	8
	0	3	4	0	6	0	0	0

依据 eh 出现的几率很小，但 he 出现的可能性较多的特点，估计 $R\leftrightarrow h$.

(5)　R 和其它字母的前后缀数目统计如下：

↓ / →	S	P	F	N	K	J	B	O
R	0	12	0	0	0	0	0	0
	8	0	5	0	0	0	0	0

根据 th 出现较多，而 ht 出现较少的特点判定 $P\leftrightarrow t$.

(6)　依据 ea 较 ae 出现的可能要大得多，以及 ha 较 ah 多的特点，估计 $F\leftrightarrow a$.

(7)　F 和 S，P，N，K，J，B，O，R 的前后缀统计如下：

→＼↓	S	P	N	K	J	B	O	R
F	3	2	1	0	1	0	0	5
	0	4	4	0	3	0	6	0

由于 a,i,o 三母音连缀的机会甚少，有理由推断 K、B 对应于 i、o 的可能性甚大，究竟 K 对应于 i 还是 o？尚要其它的根据.

(8) 在高频率字母群中，N、R 和 s 尚未判定. 他们可能和 J、N、O 以某种方式对应.

现将 $S\leftrightarrow e$，$R\leftrightarrow h$，$P\leftrightarrow t$，$F\leftrightarrow a$ 代回密文中去. 以观察 K、B 中究竟谁对应于 i，又谁对应于 O，以及 J，N，O 和明文字母 n，r，s 的对应关系. 即从上下连缀中寻求更多的线索，帮助解决存疑之点.

```
   E Q N B J   A P R S G   F O P P S   J Y S F N   Q P R S F
①  d u r i n   g t h e l   a s t t e   n y e a r   s t h e a
   6 7 2 1 2   7       4     1         2 5     2   1

   N P K T O   S V Q N B   J A F G G   T K N H O   K T E F P
②  r t o f s   e c u r i   n g a l l   f o r m s   o f d a t
   2   1 7 1     4 7 2 1   2 7   4 4   7 1 2 6 1   1 7 6

   F B J V G   Q E B J A   E B A B P   F G O L S   S V R R F
③  a i n c     u d i n g   d i g i t   a l s   e   e c h h a
     1 2 4     7 6 1 2 7   6 1 7 1       4 1         4

   O B H L N   K U S E H   F J Y T K   G E P R S   L N B H F
④  s i m   r   o   e d m   a n y   o   l d t h e     r i m a
   1 1 6   2   1     6 6     2 5   1   4 6           2   6

   N Y N S F   O K J T K   N P R B O   R F O I S   S J P R S
⑤  r y r e a   s o   n o   r t h i s   h a s b e   e n t h e
   2 5 2       1 1 2   1   2     1 1       1 3       2
```

	F E U S J	*P K T H B*	*V N K S G*	*S V P N K*	*J B V O P*
⑥	*a d e n*	*t o 5 m i*	*c r o e l*	*e c t r o*	*n i c s t*
	6 2	1 7 6 2	4 2 1 4	4 2 1	2 1 4 1
	R S V K H	*L G S X B*	*P Y K T P*	*R S T Q J*	*V P B K J*
⑦	*h e c o m*	*l e i*	*t y o f t*	*h e f u n*	*c t i o n*
	4 1 6	4 1	5 1 7	7 7 2	4 1 1 2
⑧	*O P R F P*	*V F J J K*	*W I S L S*	*N T K N H*	*S E I Y P*
	s t h a t	*c a n o*	*b e e*	*r f o r m*	*e d b y t*
	1	4 2 1	3	2 7 1 2 6	6 3 5
	R S H F V	*R B T S R*	*F O N B O*	*S J E N F*	*H F P B V*
⑨	*h e m a c*	*h i m e h*	*a s r i s*	*e n d r a*	*m a t i c*
	6 4	1 6	1 2 1 1	2 6 2	6 1 4
	F G G Y F	*O F N S O*	*Q G P K T*	*P R B O N*	*S V S J P*
⑩	*a l l y a*	*s a r e s*	*l t o*	*t h i s r*	*e c e n t*
	4 4 5	1 2 1	4 1	1 1 2	4 2
	E S U S G	*K L H S J*	*P B J P S*	*V R J K G*	*K A Y H F*
⑪	*d e e l*	*o m e n*	*t i n t e*	*c h n o l*	*o g y m a*
	5 4	1 7 6 2	1 2	4 2 1 4	1 7 5 6
	J Y K T P	*R S V B L*	*R S N O Y*	*O P S H P*	*R F P W S*
⑫	*n y o f t*	*h e c i*	*h e r s y*	*s t e m t*	*h a t e*
	2 5 1 7	4 1	2 1 5	1 6	
	N S K J V	*S V K J O*	*B E S N S*	*E O S V Q*	*N S F N S*
⑬	*r e o n c*	*e c o n s*	*i d e r e*	*d s e c u*	*r e a r e*
	2 1 2 4	4 1 2 1	1 6 2	6 1 4 7	2 2
	J K W I N	*S F D F I*	*G S*		

⑭ *no* *br* *ea* *ab* *le*

21 32 3 4

(9) 请看⑤段，若令

$B\leftrightarrow i$，$O\leftrightarrow S$

出现 this has，提供了一种可能，现将 $B\leftrightarrow i$，$K\leftrightarrow o$，$o\leftrightarrow s$ 代入密文，希望得到更多的信息支持这个判断. 在下面用 $\underset{1}{i}$，$\underset{1}{s}$，$\underset{1}{o}$ 以示这过程.

(10) 从代入后的⑦有

$\underset{t}{P}\ \ \underset{i}{B}\ \ \underset{o}{K}\ \ J$

故推断 $J\leftrightarrow n$，代入密文用 $\underset{2}{n}$ 表示这过程. 同时必然的一个结果是 $N\leftrightarrow \underset{2}{r}$，到此高频字母对应关系初步告一阶段.

(11) 从代入后的⑤可知 $I\leftrightarrow b$ 是合理的推断. 用之代入密文，用 $\underset{3}{b}$ 表之.

(12) 从代入后的⑥推测

$G\leftrightarrow l$，$V\leftrightarrow c$

以之代入密文，并用 $\underset{4}{l}$，$\underset{4}{c}$ 表示这过程.

(13) 从⑤可见 $Y\leftrightarrow \underset{5}{y}$，代入密文，表以 $\underset{5}{y}$，当然有 $Z\leftrightarrow z$.

(14) 从⑨的上下文来看 $E\leftrightarrow \underset{6}{d}$，$H\leftrightarrow \underset{6}{m}$.

(15) 从②可知 $T\leftrightarrow \underset{7}{f}$，$Q\leftrightarrow \underset{7}{u}$，$A\leftrightarrow \underset{7}{g}$.

现将已得到的对应关系列表于下：

a	b	c	d	e	f	g	h	i	j*	k*	l	m	n	o*	p*	q*	r	s	t	u	v*	w*	x*	y	z
F	I	V	E	S	T	A	R	B	C	D	G	H	J	K	L	M	N	O	P	Q	U	W	X	Y	Z

不难发现密钥词组为 *FIVE STARS*，未确定的对应关系到此全部解决，即上面打 * 的部份，如：

$j^{*}\leftrightarrow C$，　$k^{*}\leftrightarrow D$

等等.

破译到此全部结束，最终的明文是：

"during the last ten years the art of securing all forms of data including digital speech has improved many fold the primary reason for this has been the advent of microelectronics the complexity of the functions that can now be performed by the machine has risen dramatically as a result of this recent development in technology many of the cipher system that were once considered secure are now breakable."

明文的意思是"在过去的十年里，包括数字语音在内的各种数据保密技术成倍地增长，主要原因在于微电子技术的出现.它使得由机器来实现的功能明显地越来越复杂. 技术上的最新结果使得曾被认为安全的许多密码系统，现在也可以攻破."

这个密码的密钥词组是FIVE STARS.

§4 维吉尼亚(Vigenere)密码及对它的分析

(1) 令英文字母a, b, …, z对应于从00到25的整数，即：

a:00, b:01, c:02, d:03, e:04, f:05, g:06
h:07, i:08, j:09, k:10, l:11, m:12, n:13
o:14, p:15, q:16, r:17, s:18, t:19, u:20
v:21, w:22, x:23, y:24, z:25,

设密钥$k=k_1k_2\cdots k_n$. 维吉尼亚加密算法如下：

设明文是n个字母组成的字符串即：

$$m = m_1m_2\cdots m_n$$
$$E(m) = C = c_1c_2\cdots c_n$$

其中 $c_i = m_i + k_i \pmod{26}$，$i = 1, 2, \cdots, n$

例 m=data security，k=star.

首先将明文分成每段有四个字符的串.

data　secu　rity

每段用 star 加密得密文：

VTTR　KXCL　JBTR

表 1.4.1 是维吉尼亚方阵，可以利用它来加密和解密. 例如利用 *star* 对 *data* 加密得 *VTTR*，第一个 *V* 是在 *s* 行 *d* 列上找到，第二个 *T* 是在 *t* 行 *a* 列上找到，其余依此类推.

表 1.4.1　　维吉尼亚方阵

明文：	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q

s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

类似可以构造鲍福德(Beaufort)加密算法如下：

$$E(m) = c = c_1c_2\cdots c_n$$

$$c_i = k_i - m_i \pmod{26},\ i = 1,2,\cdots,n$$

鲍福德方阵见表 1.4.2.

表 1.4.2　鲍福德方阵

明文：	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*
a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
b	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B
c	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C
d	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D
e	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E
f	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F
g	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G
h	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H
i	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I
j	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J
k	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K
l	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L
m	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M
n	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N
o	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O
p	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P

q P O N M L K J I H G F E D C B A Z Y X W V U T S R Q
r Q P O N M L K J I H G F E D C B A Z Y X W V U T S R
s R Q P O N M L K J I H G F E D C B A Z Y X W V U T S
t S R Q P O N M L K J I H G F E D C B A Z Y X W V U T
u T S R Q P O N M L K J I H G F E D C B A Z Y X W V U
v U T S R Q P O N M L K J I H G F E D C B A Z Y X W V
w V U T S R Q P O N M L K J I H G F E D C B A Z Y X W
x W V U T S R Q P O N M L K J I H G F E D C B A Z Y X
y X W V U T S R Q P O N M L K J I H G F E D C B A Z Y
z Y X W V U T S R Q P O N M L K J I H G F E D C B A Z

(2) 对维吉尼亚密码的分析.

若随机地产生英文字母，并由此构成序列，任一特定的字母被选上的概率为1/26. 若两个这样的序列并列，在某一特定的位置上出现相同的字母的概率为：

$$26 \times \left(\frac{1}{26}\right)^2 = 1/26 = 0.0385$$

若任选两段英文并列，在特定的位置上同时出现字母 a 的概率为：

$$(0.0856)^2 = 0.0073274$$

同时出现字母 b 的概率为：

$$(0.0139)^2 = 0.0001932$$

其余依此类推. 令：

$$P = \sum_{i=a}^{z} p_i^2 = 0.0687$$

$$0.0687/0.0385 = 1.784$$

这说明任选两段英文并列，在特定位置上出现相同字母的概率是随机产生的两段字母序列的 1.784 倍.

如果对某一密文出现的字母频率进行统计得：$f_a, f_b, \cdots, f_z$.

$$f_a + f_b + \cdots + f_z = 1$$

令

$$k^2 = \sum_{\xi=a}^{z}\left(f_\xi - \frac{1}{26}\right)^2$$

$$= \sum_{\xi=a}^{z} f_\xi^2 - \frac{2}{26}\sum_{\xi=a}^{z} f_\xi + 1/26$$

$$= \sum_{\xi=a}^{z} f_\xi^2 - \frac{2}{26}\sum_{\xi=a}^{z} f_\xi + \frac{1}{26}$$

$$= \sum_{\xi=a}^{z} f_\xi^2 - \frac{1}{26}$$

若 $f_a = f_b = \cdots = f_z = \frac{1}{26}$，则 $k = 0$.

k^2 的值越大，表示 f_a，f_b，$\cdots$，f_z 起伏越大.

假定 $c_1c_2\cdots c_n$ 是一段密文，其中出现 a 的次数为 n_a，b 出现的次数为 n_b，等等. n_a 个字符 a 可构成 $\frac{1}{2}n_a(n_a - 1)$ 个由字母 a 构成的字母对，其余依此类推. 这样 c_1，c_2，$\cdots$，c_n 中由相同的字母构成的字母对数目为：

$$\frac{1}{2}\sum_{\xi=a}^{z} n_\xi(n_\xi - 1)$$

任意两个位置上有相同字母的概率为：

$$\mathrm{IC} = \sum_{\xi=a}^{z} n_\xi(n_\xi - 1)/n(n-1)$$

IC 是 Index of Coincidence 的缩写，称之为重合指数，表示任意两个位置上有相同字母的概率.

令 $k = k_1k_2\cdots k_m$ 是维吉尼亚密码的密钥，假定由 m 个不同的字母组成，可将密文分成 m 行使每行都是由单表置换加密的结果. 设 $c = c_1c_2\cdots c_mc_{m+1}\cdots c_{2m}c_{2m+1}\cdots$，则 m 行可如下构成：
设 $n = l \cdot m$

$$
\begin{array}{l}
c_1c_{m+1}\cdots c_{(l-1)m+1} \\
c_2c_{m+2}\cdots c_{(l-1)m+2} \\
\vdots \quad \vdots \qquad \vdots \\
c_m\ c_{2m}\cdots c_{lm}
\end{array}
$$

则 $c_1c_{m+1}\cdots c_{(l-1)m+1}$，$c_2c_{m+2}\cdots c_{(l-1)m+2}$，$\cdots$，$c_mc_{2m}\cdots c_{lm}$都是单表置换的结果. 也就是说每一行中任意两位有相同字母的概率约为 0.0687，从不同的行任取两位有相同字母的概率约 0.0385，即将它们看作是随机序列中的两位.

在选定了第一位置后，从相同行中选第二个位置的方案数为 $\frac{1}{2}n\left(\frac{n}{m}-1\right)$，在不同行中选第二个位置的方案数为 $\frac{1}{2}n\left(n-\frac{n}{m}\right)$，故

$$
\begin{aligned}
\frac{1}{2}n(n-1)\mathrm{IC} = \frac{1}{2}n\left(n-\frac{n}{m}\right)\times 0.0385 \\
+\frac{1}{2}n\left(\frac{n}{m}-1\right)\times 0.0687
\end{aligned}
$$

$$
\therefore (n-1)\mathrm{IC} = \frac{n}{m}(0.0687-0.0385)+0.0385n-0.0687
$$

$$
\therefore m = \frac{0.0302}{(n-1)\mathrm{IC}-0.0385n+0.0687}
$$

这个公式可用来估计密钥 k 的长度. 这是破译维吉尼亚密码所必需的.

(3) 卡席斯基(Kasiski)分析.

对维吉尼亚密码的分析，首先解决出密钥的长度. 下面介绍一种卡席斯基分析方法. 明文中相同的两个字母的密文可能是不一样的. 然而英文中，*th*，*ing*，*ed*，*tion* 等字符出现较多，只要它们之间的距离正好是密钥长度的倍数，则情况显然又是另外一回事. 卡席斯基分析便是从这入手，分析密文中重复出

现的字符串，并求出它们的距离进行分析，举例如下：

已知密文如下：

U F Q U I U D W F H G L Z A R I H W L L W Y Y F
S Y Y Q A T J P F K M U X S S W W C S V F A E V
W W G Q C M V V S W F K U T B L L G Z F V I T Y
O E I P K S J W G G S J E P N S U E T P T M P O
P H Z S F D C X E P L Z Q W K D W F X W T H A S
P W I U O V S S S F K W W L C C E Z W E U E H G
V G L R L L Q W O F K W L U W S H E V W S T T U
A R C W H W B V T G N I T J R W W K C O T F G M
I L R Q E S K W G Y H A E N D I U L K D H Z I Q
A S F M P R G W R V P B U I Q Q D S V M P F Z M
V E G E E P F O D J Q C H Z I U Z Z M X K Z B G
J O T Z A X C C M U M R S S J W

对 280 个字符进行统计得 A 为 9 个，B 为 4 个，……，Y 共 6 个，Z 有 12 个．IC 计算得 0.0431．

作 Kasiski 试验．以 UI 为例在第 4 位第 1 次出现，但在第 229 位第 2 次出现，其间距离为 225．IU 在第 5 位第 1 次出现，第 123 和 208，255 次重复出现，其间距离分别为 118，85，47，……但

$$225=5\times5\times3\times3$$

$$85=5\times17$$

……

试验结果发现 2 和 5 的因数出现的频率较高．

密钥长为 2 的可能性不大，所以密钥长度可能为 5．现将密文分 5 行，并求其重合指数 IC 如下：

U U G I W Y J U W A G V U G T F G P T O F P K W P V K C
U G G W H T C V T K G Q G N K Q P V Q M V P Q U K O C R

1：IC＝0.0623

F D L H Y Y P X C E Q S T Z Y A G N P P D L D T W S W E
E L W L E T W T J C M E Y D D A R P Q P E F C Z Z T C S

2：IC＝0.0506

Q W Z W Y Q F S S V C W B F O S S S T H C Z W H I S W Z
H R O U V U H G R O I S H I H S G B D F G O H Z B Z M S

3：IC＝0.0649

U F A L F A K S V W M F L V E J J U M Z X Q F A U S L W
G L F W W A W N W T L K A U Z F W U S Z E D Z M G A U J

4：IC＝0.0617

I H R L S T M W F W V K L I I W E E P S E W X S O F C E
V L K S S R B I W P R W E L I M R I V M E J I X J X M W

5：IC＝0.0617

可见除第 2 行外，其余行的 IC 都在 0.60 以上，有力的说明它们可能分别是单表置换的结果.

上面的 5 行假定它们各自通过

$$c \equiv m + k_i \pmod{26}$$

变换得到的，不同的是密钥 k_i. 假定第 i 行的密钥为 k_i，第 j 行的密钥为 k_j，令 $\delta_{i,j}=k_j-k_i$，后面将设法求出 $\delta_{i,j}$，目的在于将它们联接起来，使之成为同一密钥 k 加密的单一的密文.

设第 i 行的字符数为 n，而第 j 行的字符数为 $\bar{n}$，同理第 i 行的字符 A 的数目设为 n_A，第 j 行字符 A 的数目为 $\bar{n}_A$，其它依此类推. 将两行合并起来的重合指数为

$$\mathrm{IC} = \sum_{\xi=A}^{Z}(n_\xi+\bar{n}_\xi)(n_\xi+\bar{n}_\xi-1)/[(n+\bar{n})(n+\bar{n}-1)]$$

但 $$\sum_{\xi=A}^{Z}(n_\xi+\bar{n}_\xi)(n_\xi+\bar{n}_\xi-1)$$

$$= \sum_{\xi=A}^{Z} n_\xi^2 + \sum_{\xi=A}^{Z} \bar{n}_\xi^2 + 2\sum_{\xi=A}^{Z} n_\xi \bar{n}_\xi - \sum_{\xi=A}^{Z} n_\xi - \sum_{\xi=A}^{Z} \bar{n}_\xi$$

$$= \sum_{\xi=A}^{Z} n_\xi^2 + \sum_{\xi=A}^{Z} \bar{n}_\xi^2 + 2\sum_{\xi=A}^{Z} n_\xi \bar{n}_\xi - n - \bar{n}$$

所以可对第 i 行和第 j 行进行试配合时，可将第 i 行的每一元素作加 k（mod 26）的运算．这里 $k=0, 1, 2, \cdots, 25$. 观察使 IC 达到最大的 k 的值，其实也就是使 $\sum_{\xi=A}^{Z} n_\xi \bar{n}_\xi$ 达到最大的 k 值 k^*.

例如，第 1 行

U U G I W Y J U W A G V U G T F G P T …

当 $k=2$ 时，便得相应的

W W I K Y A L W A C I X W I V H I R V …

现将第 i 行每位后推 k 位得到的新序列和第 j 行联合，计算 $\sum_{\xi=A}^{Z} n_\xi \bar{n}_\xi$，并将结果列表如下：

$i=1$	104	101	112	127	106	131	104	109	131	209*	144	98	103	
$j=2$	140	112	108	125	103	107	108	124	101	141	139	139	110	$k^*=9$
$i=1$	112	152	142	122	100	112	121	93	139	91	110	128	226*	
$j=3$	129	102	91	124	94	121	120	90	90	116	148	129	134	$k^*=12$
$i=1$	133	125	107	134	125	159	151	108	70	100	159	126	126	
$j=4$	76	114	143	209*	117	92	90	135	95	95	90	116	141	$k^*=16$
$i=1$	94	131	218*	132	74	90	131	105	118	105	92	107	146	
$j=5$	115	113	154	161	125	104	94	111	112	122	107	164	111	$k^*=2$
$i=2$	107	109	121	214*	109	115	103	127	84	125	116	100	99	
$j=3$	96	165	134	143	92	153	107	118	114	147	106	95	137	$k^*=3$
$i=2$	127	148	125	131	99	103	121	194*	114	124	93	115	89	
$j=4$	112	104	117	134	106	138	125	143	105	155	119	118	77	$k^*=7$
$i=2$	132	99	103	120	130	122	152	129	129	100	96	123	136	
$j=5$	105	108	158	119	91	147	206*	110	86	103	116	98	123	$k^*=19$

$i=3$	129	98	130	119	217*	113	125	94	134	88	95	95	118	
$j=4$	162	122	138	85	133	214	164	108	114	69	123	154	95	$k^*=4$
$i=3$	142	137	99	141	152	145	94	77	97	122	120	97	155	
$j=5$	118	108	134	209*	109	94	111	128	88	115	150	95	99	$k^*=16$
$i=4$	140	113	113	110	108	111	104	93	141	117	120	128	188*	
$j=5$	125	107	93	120	133	156	112	58	102	149	146	104	145	$k^*=12$

其中 k^* 是 k 最大值所在位置．可见词组密钥的第 1 个字母和第 2 个字母间距离为 9，第 2 个到第 3 个间距离为 3，…

根据以上的分析，可得可能的密码有

AJMQC	BKNRD	CLOSE	DMPTE	ENQUC
FORVH	GPSWI	HQTXJ	IRUYK	JSVZL
KTWAM	LUXBN	MVYCO	NWZDP	OXAEQ
PYBFR	QZCGS	RADHT	SBEIU	TCFJV
UDGKW	VEHLX	WFIMY	XGJNZ	YHKOA
		ZILPB		

试将 5 段密文联接得密文(C^*)

U W E E G U U K P F G C N K P I Y K V J W P M P Q Y P E
K R J G T U K U O G C U W T G F D A V J G U G H Q W T V
J K P I U K P V J G Q T F G T P C O G F R G T U G X G T
C P E G E C T G H W N O G V J Q F U Q H C P C N A U K U
K P V W K V K Q P N W E M V U G C D K N K V A C V N G C
U V V Q T G C F V J G N C P I W C I G Q H V J G Q T K I
K P C N V G Z V K U X G T A F G U K T C D N G D W V P Q
V G U U G P V K U N U W E J K U V J G Q R G P K P I U G
P V G P E G Q H R C T M G T J K V V U O C P W C N H Q T
V J G U Q N W V K Q P Q H O K N K V C T A E K R J G T U

统计 26 个英文字母出现的频数：

A	B	C	D	E	F	G	H	I	J	K
5	0	20	4	9	7	36	7	6	14	25
L	M	N	O	P	Q	R	S	T	U	V
0	3	13	5	22	16	5	0	17	23	26
W	X	Y	Z							
12	2	2	1							

相应的重合指数 IC＝0.0654，因而密文 C^* 可看作是下面明文 m 通过凯撒变换加密而成的：

Success in dealing with unknown ciphers is measured by these four things in the order named perseverance, careful methods of analysis, intuition, luck. The ability at least to read the language of the original text is very desirable but not essential. Such is the opening sentence of Parker Hitt's manuel for the Solution of Military Ciphers.

§5 不确定性的度量——熵的概念

（1）从密码学的分析可以看出，明文是毫无不确定性可言的．密文则不然，随着分析的进行，不确定性程度逐渐减小，最后完全确定．不同的密码，它们的强度也不一样．如何测度不确定性的程度呢？下面引进熵的概念，它是由商农提出的．商农的信息论是密码学的一个理论基础．

若有 k 个等概率事件，$k=1$ 时是确定性事件，它没有任何不确定的因素，但随着 k 的增加，不确定的成分便随之增大，从信息的角度来看，如果获得了确定性事件的信息，这个信息就没有价值，或者说它的信息量等于零．比如说：“我出门时看见

太阳从东方升起”. 没有人对这句话感兴趣，因为从中得不到任何新东西. 但是如果说：“我见到不明的飞行物”，则完全不一样. 所以，不确定性越大，信息量也越大，反之，则越小. 而确定性事件信息量等于零.

如何度量不确定性呢？假定用一个 k 的函数来表示它，它满足：

(a) $f(1)=0$;

(b) 若 $k_1>k_2$，则 $f(k_1)>f(k_2)$;

(c) 若 A 有 l 种等概率事件：$a_1,a_2,\cdots,a_l$，B 有 m 种等概率事件：$b_1,b_2,\cdots,b_m$，A 和 B 同时出现的事件有 ml 种，显然 $f(lm)>f(l)$，$f(lm)>f(m)$. 令：$f(lm)=f(l)+f(m)$，很自然地想起

$$f(k)=\log k=-\log\frac{1}{k}$$

本书假定对数以 2 为底，结果以“比特”为单位，当然以 e 作底或以 10 作底也是可以的.

可将这个概念推广到若有 k 个不等概率的事件 A：a_1，a_2，$\cdots$，a_n，假定它们出现的概率分别为 P_1，P_2，$\cdots$，P_k，满足

$$P_1+P_2+\cdots+P_k=1.$$

定义 1.5.1 $H(A)=-P_1\log P_1-P_2\log P_2-\cdots-P_k\log P_k$ 则称 $H(A)$ 为事件 A 的熵.

若 $P_1=P_2=\cdots=P_k=1/k$ 时，$H(A)=\log k$，和前面等概率事件的讨论是一致的.

下面用 $H(A)$ 来衡量事件 A 的不确定性.

例 对某一地区进行多年的观察，5 月份雨天的概率为 0.4，晴天的概率为 0.6，而 11 月份晴天的概率为 0.65，下雪的概率为 0.15，雨天的概率为 0.2，试问它们的不确定性哪个

较大?

令 A 为 5 月份的天气，B 为 11 月份的天气.

$$H(A) = -0.4\log 0.4 - 0.6\log 0.6 \approx 0.9694(\text{比特})$$

$$H(B) = -0.65\log 0.65 - 0.15\log 0.15 - 0.2\log 0.2 \approx 1.2772(\text{比特})$$

$\therefore \quad H(B) > H(A)$

即 11 月份天气的不确定性较大.

(2) 熵的基本性质.

(a) $\lim\limits_{P\to 0}(-P\log P) = 0$

利用罗比塔(I. Hopital) 法则可得:

$$\lim_{P\to 0}(P\ln P) = \lim_{P\to 0}\frac{\ln P}{\frac{1}{P}} = \lim_{P\to 0}\frac{\frac{1}{P}}{-\frac{1}{P^2}} = \lim_{P\to 0}(-P) = 0$$

(b) 当 $P_1 = P_2 = \cdots = P_k = \frac{1}{k}$ 时，$H(A)$ 的值达到最大. 也就是当 A 的 k 个事件为等概率时，其不确定性达到最大.

直观上也是 k 个事件为等概率时，不确定性达到最大. 问题是在 $P_1 + P_2 + \cdots + P_k = 1$ 条件下求:

$$H(A) = P_1\log\frac{1}{P_1} + P_2\log\frac{1}{P_2} + \cdots + P_k\log\frac{1}{P_k}$$

的极值.

利用拉格朗日(Lagrange) 乘数法，求:

$$H(A) + \lambda\sum_{l=1}^{k}P_l = \sum_{l=1}^{k}P_l\log\frac{1}{P_l} + \lambda\sum_{l=1}^{k}P_l$$

的无条件极值，即求:

$$\frac{\partial H}{\partial P_l} + \lambda\frac{\partial}{\partial P_l}\sum_{j=1}^{k}P_j = 0, \quad l = 1,2,\cdots,k$$

$\therefore \quad -\log(P_l + 1) + \lambda = 0, \ l = 1,2,\cdots,k$

故 $P_1=P_2=\cdots=P'_k=\frac{1}{k}$

(c) $H(P_1,P_2,\cdots,P_k)=H(P_1+P_2,P_3,\cdots,P_k)$

$$+(P_1+P_2)H\left(\frac{P_1}{P_1+P_2},\frac{P_2}{P_1+P_2}\right)$$

证：可以采取直接验证的方法证：右端=左端.

$$H(P_1+P_2,P_3,\cdots,P_k)+(P_1+P_2)H\left(\frac{P_1}{P_1+P_2},\frac{P_2}{P_1+P_2}\right)$$

$$=-(P_1+P_2)\log(P_1+P_2)-P_3\log P_3-\cdots-P_k\log P_k$$

$$-(P_1+P_2)\left[\frac{P_1}{P_1+P_2}\log\frac{P_1}{P_1+P_2}+\frac{P_2}{P_1+P_2}\log\frac{P_2}{P_1+P_2}\right]$$

$$=-(P_1+P_2)\log(P_1+P_2)-P_3\log P_3-\cdots-P_k\log P_k$$

$$-P_1\log P_1-P_2\log P_2+(P_1+P_2)\log(P_1+P_2)$$

$$=-P_1\log P_1-P_2\log P_2-\cdots-P_k\log P_k$$

还有更一般的结果可以作为推论：

$$H(P_1,P_2,\cdots,P_i,P_{i+1},\cdots,P_k)$$

$$=H(P_1+P_2+\cdots+P_i,P_{i+1},\cdots;P_k)+(P_1+P_2+\cdots+P_i)$$

$$\times H\left(\frac{P_1}{P_1+P_2+\cdots+P_i},\cdots,\frac{P_i}{P_1+P_2+\cdots+P_i}\right)$$

§6 暧昧度

设事件 A 有 a_1，a_2，…，a_l 种可能结果，它们出现的概率为 $P(a_i)$，$i=1$，2，…，l. 设事件 B 有 m 种可能结果 b_1，b_2，…b_m. 它们的概率为 $P(b_j)$. $j=1$，2，…，m. 则 A 和 B 同时出现的事件 AB 有 lm 种种可能结果. 若 A 和 B 是相互独立的，则 $P(a_ib_j)=P(a_i)P(b_j)$. 也容易证明，这时

$$H(AB) = H(A) + H(B)$$

证明如下：

$$\because \quad \sum_{i=1}^{l} P(a_i) = \sum_{j=1}^{m} P(b_j) = 1$$

$$H(AB) = -\sum_{i=1}^{l}\sum_{j=1}^{m} P(a_i b_j)\log P(a_i b_j)$$

$$= -\sum_{i=1}^{l}\sum_{j=1}^{m} P(a_i)P(b_j)[\log P(a_i) + \log P(b_j)]$$

$$= -\sum_{i=1}^{l} P(a_i)\log P(a_i)\sum_{j=1}^{m} P(b_j)$$

$$-\sum_{i=1}^{l} P(a_i)\sum_{j=1}^{m} P(b_j)\log P(b_j)$$

$$= H(A) + H(B)$$

上面这个等式是事件 A 和 B 互相独立的结果．如若不然，令：

$$P(a_i b_j) = P(a_i)P_{a_i}(b_j),\ i = 1,2,\cdots,l;\ j = 1,2,\cdots,m$$

$$H(AB) = -\sum_{i=1}^{l}\sum_{j=1}^{m} P(a_i)P_{a_i}(b_j)[\log P(a_i) + \log P_{a_i}(b_j)]$$

$$= -\sum_{i=1}^{l} P(a_i)\left(\sum_{j=1}^{m} P_{a_i}(b_j)\right)\log P(a_i)$$

$$-\sum_{i=1}^{l} P(a_i)\sum_{j=1}^{m} P_{a_i}(b_j)\log P_{a_i}(b_j)$$

$$\text{但} \quad \sum_{j=1}^{m} P_{a_i}(b_j) = \sum_{j=1}^{m}\frac{P(a_i b_j)}{P(a_i)} = \frac{1}{P(a_i)}\sum_{j=1}^{m} P(a_i b_j) = 1$$

$$\therefore \quad H(AB) = H(A) + H_A(B)$$

$$\text{其中} \quad H_A(B) = \sum_{i=1}^{l} P(a_i)H_{a_i}(B)$$

$$H_{a_i}(B) = -\sum_{j=1}^{m} P_{a_i}(b_j)\log P_{a_i}(b_j)$$

$H_A(B)$ 称为给定了 A 的条件下 B 的熵，或暧昧度.

若 M 为明文空间，C 为密文空间. 在截取密文 $c\in C$ 的条件下，明文 M 的条件熵为

$$H_c(M) = -\sum_{m\in M} P_c(m)\log P_c(m)$$

现在取得一段凯撒密码的密文：

htsxnijwfgqj

它的明文可能有如下 26 个

h	*t*	*s*	*x*	*n*	*i*	*j*	*w*	*f*	*g*	*q*	*j*
i	*u*	*t*	*y*	*o*	*j*	*k*	*x*	*g*	*h*	*r*	*k*
j	*v*	*u*	*z*	*p*	*k*	*l*	*y*	*h*	*i*	*s*	*l*
k	*w*	*v*	*a*	*q*	*l*	*m*	*z*	*i*	*j*	*t*	*m*
l	*x*	*w*	*b*	*r*	*m*	*n*	*a*	*j*	*k*	*u*	*n*
m	*y*	*x*	*c*	*s*	*n*	*o*	*b*	*k*	*l*	*v*	*o*
n	*z*	*y*	*d*	*t*	*o*	*p*	*c*	*l*	*m*	*w*	*p*
o	*a*	*z*	*e*	*u*	*p*	*q*	*d*	*m*	*n*	*x*	*q*
p	*b*	*a*	*f*	*v*	*q*	*r*	*e*	*n*	*o*	*y*	*r*
q	*c*	*b*	*g*	*w*	*r*	*s*	*f*	*o*	*p*	*z*	*s*
r	*d*	*c*	*h*	*x*	*s*	*t*	*g*	*p*	*q*	*a*	*t*
s	*e*	*d*	*i*	*y*	*t*	*u*	*h*	*q*	*r*	*b*	*u*
t	*f*	*e*	*j*	*z*	*u*	*v*	*i*	*r*	*s*	*c*	*v*
u	*g*	*f*	*k*	*a*	*v*	*w*	*j*	*s*	*t*	*d*	*w*
v	*h*	*g*	*l*	*b*	*w*	*x*	*k*	*t*	*u*	*e*	*x*
w	*i*	*h*	*m*	*c*	*x*	*y*	*l*	*u*	*v*	*f*	*y*
x	*j*	*i*	*n*	*d*	*y*	*z*	*m*	*v*	*w*	*g*	*z*
y	*k*	*j*	*o*	*e*	*z*	*a*	*n*	*w*	*x*	*h*	*a*
z	*l*	*k*	*p*	*f*	*a*	*b*	*o*	*x*	*y*	*i*	*b*

a	*m*	*l*	*q*	*g*	*b*	*c*	*p*	*y*	*z*	*j*	*c*
b	*n*	*m*	*l*	*h*	*c*	*d*	*q*	*z*	*a*	*k*	*d*
c	*o*	*n*	*s*	*i*	*d*	*e*	*r*	*a*	*b*	*l*	*e*
d	*p*	*o*	*t*	*j*	*e*	*f*	*s*	*b*	*c*	*m*	*f*
e	*q*	*p*	*u*	*k*	*f*	*g*	*t*	*c*	*d*	*n*	*g*
f	*r*	*q*	*v*	*l*	*g*	*h*	*n*	*d*	*e*	*o*	*h*
g	*s*	*r*	*w*	*m*	*h*	*i*	*v*	*e*	*f*	*p*	*i*

若取一个字母进行判断，无疑 e 的概率最大，这时它的条件熵用 $H_C(M,1)$ 表之，有：

$$H_c(M,1) = -\sum_{\xi=a}^{z} P(\xi)\log P(\xi) = 4.125(\text{比特})$$

若取两个字母进行分析，设 $m=\xi\eta$，但

$$P(\xi\eta) = P(\xi)P_\xi(\eta)$$

其中 $P_\xi(\eta)$ 为字母 ξ 后缀 η 的概率．如 $P(a)=0.0856$，$P(b)=0.0139$ 等一样，$P_\xi(\eta)$ 有统计结果（从略）．计算结果见表：

表 1.7.1

$m=\alpha\beta$	$P(\alpha)P_\alpha(\beta)$	$P_c(m)$
am	0.00241	0.07341
bn	0	0
co	0.00637	0.19441
dp	0	0
lq	0.00044	0.01353
fr	0.00351	0.10717
gs	0.00051	0.01555
ht	0.00123	0.03755

续表

$m=\alpha\beta$	$P(\alpha)P_e(\beta)$	$P_c(m)$
iu	0.00007	0.00211
jv	0	0
ku	0	0
lx	0	0
my	0.00048	0.01459
nz	0.00003	0.00086
oa	0.00068	0.02068
pb	0	0
qc	0	0
rd	0.00191	0.05827
se	0.01090	0.33255
tf	0.00007	0.00223
ug	0.00095	0.02903
vh	0	0
wi	0.00314	0.09568
xj	0	0
yk	0.00007	0.00207
zl	0.00001	0.00031

可算得

$$H_c(M,2) = -\sum_{m\in c_2} P_c(m)\log P_c(m)$$
$$= 2.9497(\text{比特})$$

类似可得

$$H_c(M,3) = 1.2115(\text{比特})$$
$$H_c(M,4) = 0.9356(\text{比特})$$
$$H_c(M,5) = 0(\text{比特})$$

即截获 5 个字符时便可确定明文，现将 $n=3$，4，5 的计算结果列为表 1.7.2. 括号里的数便是 $P_c(m)$.

表 1.7.2

m	$n=3$	$n=4$	$n=5$
a m l q g	0.00002(0.00717)	0	0
b n m r h	0	0	0
c o n s i	0.00139(0.49821)	0.00010(0.29412)	0.00001(1.00000)
d p o t j	0	0	0
p q p u k	0	0	0
f r q v l	0	0	0
g s r w m	0	0	0
h t s x n	0.00005(0.01792)	0	0
i u t y o	0.00001(0.00358)	0	0
j v u z p	0	0	0
k w v a q	0	0	0
l x w b r	0	0	0
m y x c s	0	0	0
n z y d t	0	0	0
o a z l u	0	0	0
p b a f v	0	0	0
q c b g w	0	0	0
r d c h x	0.00001(0.00358)	0.00001(0.02941)	0
s e d i y	0.00130(0.46595)	0.00023(0.67847)	0
t f e j z	0.00001(0.00358)	0	0
u g f k a	0	0	0
v h g l b	0	0	0
w i h m c	0	0	0
x j i n d	0	0	0
y k j o e	0	0	0
x l k p f	0	0	0

从以上可知，随着 n 的增加不确定性随之减少. $n=1$ 时 e

的概率最大. $n=2$ 时，暧昧度降至 2.9497 比特. 以 *se* 的出现概率为最大. $n=3$ 时 *con* 和 *sed* 出现的概率分别为 0.49821 和 0.46595. $n=4$ 时情况仍不明朗. $n=5$ 时 $H_C(M,5)=0$，不确定性消失. 即明文应为 considerable.

§7 商农(Shannon)理论

(1) 随机密码是假定：

(a) 密钥空间所有密钥均匀分布，即每个密钥概率相等.

(b) 长度为 n 的字符串由两部分构成，一部分是有意义的，另一部分是无意义的.

(c) 有意义部分的字符串出现的概率也是相等的.

英文字母 26 个，由它们构成 n 位字符串的数目为 $26^n=2^{rn}$，$r=\log_2 26=4.7004$，其中有意义的一类其数目设为 $2^{\alpha_n n}$，则传输长度为 n 的字符串为有意义明文的概率为：

$$
\begin{aligned}
P &= 2^{\alpha_n n}/2^{rn} \\
&= 2^{-(r-\alpha_n)n} \\
&= 2^{-d_n n}
\end{aligned}
$$

其中 $d_n=\gamma-\alpha_n$

称为语言的冗余度.

随机密码假设 n 位字符串中有意义的部分出现的概率是相等的. 所以，任取密文 c，随机地选择一密钥对 c 解密，得到有意义明文的概率为 $2^{-d_n n}$. 因此，冗余度愈大获得有意义的明文的概率愈小.

(2) 唯一解码量.

商农的贡献在于他提出了以冗余度作为密码分析的基础.

从此，进一步对具有某种冗余度的明文定量给出破译密文所需的字母数目，他称之为唯一解码量. 假定密钥空间 $K=\{k_1,k_2,\cdots,kN_k\}$ 中每一个密钥都是等概率的，这样每一个密钥的概率为：

$$P=1/N_k$$

$$H(K)=-N_k\frac{1}{N_k}\log_2\frac{1}{N_k}=\log_2 N_k$$

所以，用所有的密钥进行脱密可能得到有意义明文的期望值等于 $2^{H(K)}2^{-d_n n}=2^{H(K)-d_n n}$. 若 $H(K)$ 比 $d_n n$ 大，则获得有意义译文的数目也多. 当 $H(K)=d_n n$ 时，有意义的译文正好只有一个，这个 n 就称之为唯一解码量，用 Ud (unicity distance) 表示它，即：

$$\mathrm{U}d\cdot d_n=H(K)$$

Ud 给出了破译密码所需的最少密文字符数，也就是确定密钥所需的最少字符数.

例如对于词组密钥密码，密钥的数目为 26!

$$\begin{aligned}H(K)&=-(26!)\frac{1}{26!}\log_2\left(\frac{1}{26!}\right)\\&=\log_2(26!)\\&=88.382(\text{比特})\end{aligned}$$

$$|A|=26,\ r=\log_2 26=4.7004$$

至于有意义的明文数目，有多种不同的估计. 先介绍其中最简单的一种，当明文的长度 n 充分大时，26 个英文字母出现的数目设为 n_a，n_b，$\cdots$，n_z. 若有意义的明文的概率为 P，则：

$$P=P_a^{n_a}P_b^{n_b}\cdots P_z^{n_z}.$$

其中 P_a，P_b，$\cdots$，P_z分别是英文字母 a、b，$\cdots$，z 出现的概率，而且有：

$$n_a=nP_a,\ n_b=nP_b,\ \cdots,\ n_z=nP_z.$$

$$\therefore \quad P = (P_a^{p_a} P_b^{p_b} \cdots P_z^{p_z})^n$$

令长度为 n 的有意义明文数目为 S，根据假定它们出现的概率相等，即：

$$P = \frac{1}{S},\ S = \frac{1}{P}$$

$$\log_2 S = -\log_2 P = -n(P_a \log_2 P_a + P_b \log_2 P_b + \cdots + P_z \log_2 P_z)$$

依据统计结果：

$$\sum_{\xi = a}^{z} P_\xi \log_2 P_\xi = -4.192$$

$$\therefore \quad S = 2^{4.192n}$$

$$a_n = 4.192$$

但 $\gamma_n = 4.7004$

$$d_n = r_n - a_n = 4.7004 - 4.192 = 0.5084$$

$$\mathrm{Ud} = H(K)/d_n = 88.382/0.5084 \approx 174$$

也就是说对于词组密钥密码系统来说，唯一解码量为 174 个字符.

唯一解码量的估计归根到底和有意义明文数目的估计相关. 上面的例子是假定：

$$P = P_a^{n_a} P_b^{n_b} \cdots P_z^{n_z} = (P_a^{p_a} P_b^{p_b} \cdots P_z^{p_z})^n$$

没有考虑到前后缀的关系. 比如对 2-字母组，

$$m = m_1 m_2 m_3 m_4 \cdots m_{2n-1} m_{2n}$$

出现概率为

$$P(m) = P(m_1 m_2) P(m_3 m_4) \cdots P(m_{2n-1} m_{2n})$$

唯一解码量只是理论地估计到至少需要多少个密文字母可以确定明文，但没有解决如何确定，需要多少计算量.

(3) 完全保密.

假定 $P(m)$ 表示明文 m 被发送的概率，同样 $P(c)$ 为收到密文 c 的概率，$P_m(c)$ 为发送明文 m，收到密文 c 的概率，依此

同样解释 $P_c(m)$．根据概率乘法定理有：

$$P_c(m)P(c)=P_m(c)P(m)$$

所谓的完全保密指的是满足等式

$$P_c(m)=P(m)$$

的密码系统，直观上表明截获密文 c 对确定明文无帮助，不难推知：

$$P_c(m)P(c)=P(m)P(c)=P_m(c)P(m)$$

$$\therefore\quad P(c)=P_m(c)$$

实际上 $P(c)=P_m(c)$ 是完全保密的充要条件.

什么样的密码系统才是完全保密的？可以从 $P_m(c)=P(c)$ 和 $P_c(m)=P(m)$ 这两个条件来观察.

假如明文空间 $M=\{m_1,m_2,\cdots,m_p\}$，密文空间 $C=\{c_1,c_2,\cdots,c_q\}$，密钥空间 $K=\{k_1,k_2,\cdots,k_r\}$，给定 $c\in C$，但

$$P_c(m)=P(m)$$

说明 C 空间里的任何一个密文，都可以在密钥空间 K 里找到一个密钥将它解密成明文 m，而且机会均等. 同样从 $P_m(c)=P(c)$ 可说明任何一个明文 $m\in M$，都可以找一密钥 k 将它加密成密文 c，而且机会也均等，如若不然，若 $m\in M$，不存在 $k\in K$，将 m 加密成 c. 已知 c，便可断定 $P_c(m)=0$，和 $P_c(m)=P(m)$ 的假定矛盾. 这样密文 c 对断定 m 不是没帮助，恰恰相反，破译者便排除 m 的可能性，提高了破译可能性. 换一句话说，完全密码系统要求密钥量至少应该和明文一样多.

§8 数据加密标准(DES)

DES 是 Data Encryption Standard 的缩写，意即数据加密

标准. 现在美国用的数据加密标准（简称为DES）是由IBM公司研制的. 1977年美国国家标准局批准它供非机密机构保密通信使用，DES的公布在密码学发展史上是件大事. 在这以前保密通信双方使用的密码系统都是由通信双方秘密约定，通信密码的标准化是我们这个信息时代的需要，它带来的好处是显而易见的，但必须完全可靠，能经受住强有力的攻击.

DES是典型的传统密码体制，它利用传统的换位和置换等加密方法. 现在介绍算法如下：

假定信息空间都是由{0，1}组成的字符串，信息被分成64比特的块，密钥是56比特. 经过DES加密的密文也是64比特的块. 设m是一个64比特的信息块，k为56比特的密钥，即：

$$m = m_1m_2\cdots m_{64} \quad m_i = 0,1, \quad i = 1,2,\cdots,64$$

$$k = k_1k_2\cdots k_{64} \quad k_i = 0,1, \quad i = 1,2,\cdots,64$$

其中k_8，k_{16}，k_{24}，k_{32}，k_{40}，k_{48}，k_{56}，k_{64}是奇偶校验位，真正起作用的仅56位.

$$\mathrm{DES}(m) = \mathrm{IP}^{-1} \circ T_{16} \circ T_{15} \circ T_{14} \circ \cdots \circ T_2 \circ T_1 \circ \mathrm{IP}(m) \tag{1.8.1}$$

IP是初始置换，IP^{-1}是IP的逆，T_i，$i=1$，2，…，16是一系列的变换，分别叙述如下：

(1) 初始置换IP（表1.8.1).

这说明若$m=m_1m_2\cdots m_{64}$

$$\mathrm{IP}(m) = m_{58}m_{50}m_{42}m_{34}\cdots m_{23}m_{15}m_7$$

同样理解IP^{-1}，IP中的第一位是58，请注意IP^{-1}中的第58位正好是1，也就是说在IP的置换下第58位换为第1位，不言而喻，在IP^{-1}的置换下，应将第1位换回第58位，余此类推.

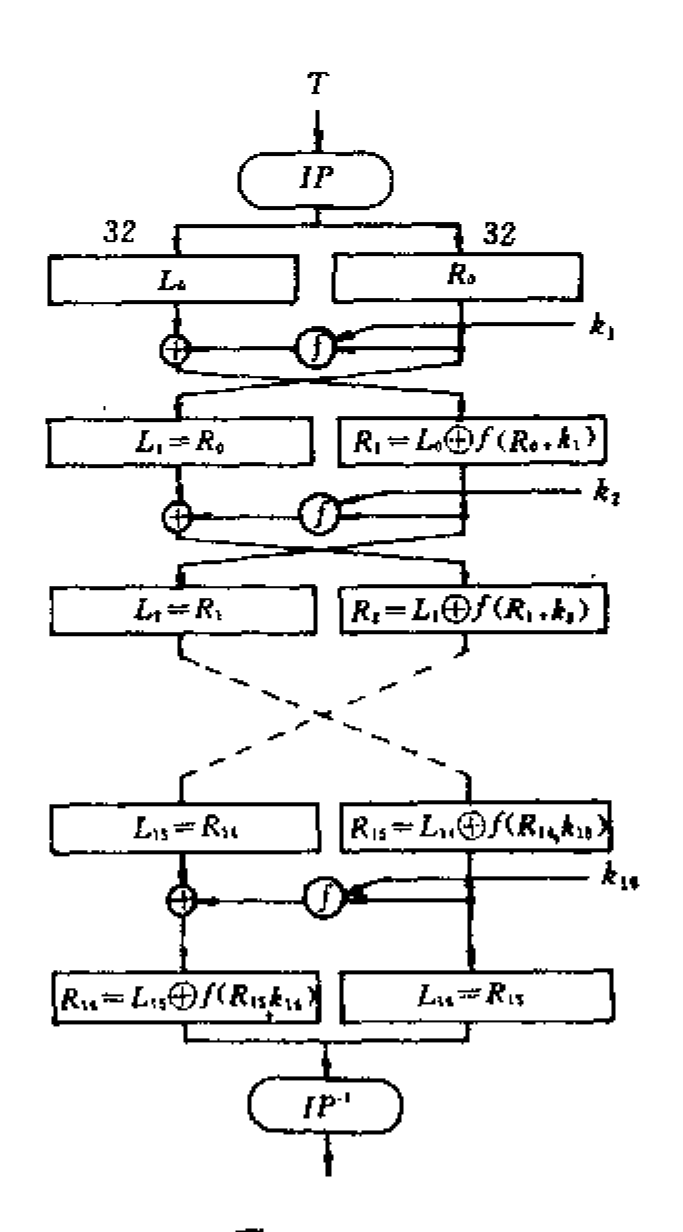


图 1.8.1

表 1.8.1 IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

表 1.8.2 IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

(2) 迭代过程 T.

设第 i 次迭代 (T_i) 的输入为 $L_{i-1}R_{i-1}$，其中 L_{i-1}，R_{i-1}分别是左半部 32 比特和右半部分 32 比特，则第 i 次迭代的输出（也就是第 $i+1$ 次迭代的输入）为：

$$L_i = R_{i-1},\ R_i = L_{i-1} \oplus f(R_{i-1},k_i)$$

其中⊕是按位的模 2 加，即：

$$0\oplus 0=0,\ 0\oplus 1=1\oplus 0=1,\ 1\oplus 1=0$$

k_i是由 56 比特的密钥 k 确定的 48 比特密钥，f 是将 32 比特的符号串映象为 32 比特的符号串. 具体算法见图 1.8.2.

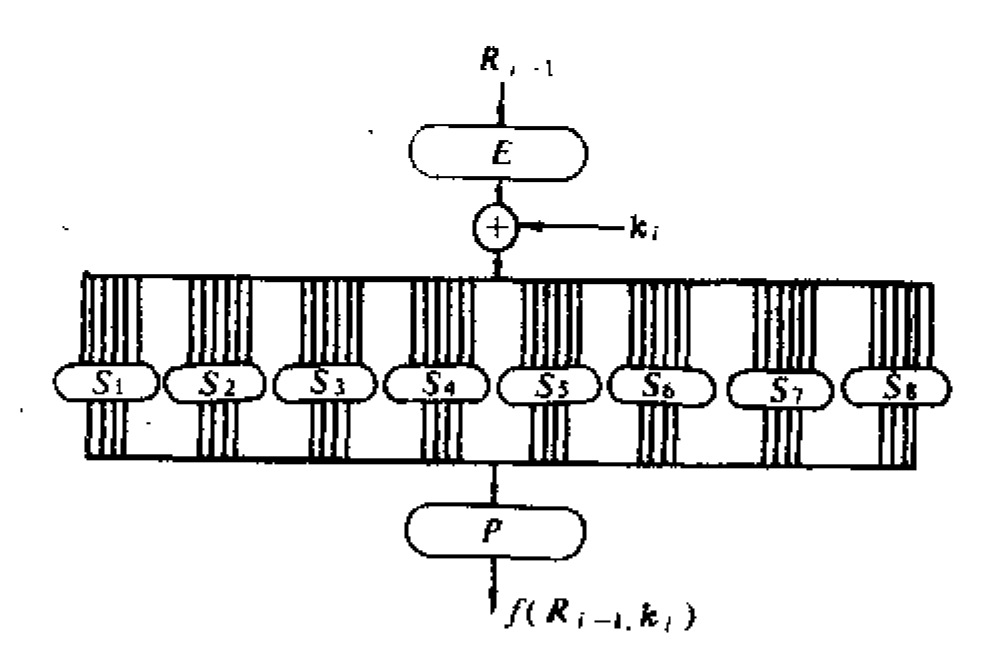


图 1.8.2

$f(R,k)$ 功能的第一步是由 E 将 32 比特的输入转为 48 比特，与 k_i 作⊕运算后分成 8 组，每组 6 比特，分别通过 S_1，S_2，…，S_8，输出还是 8 组 32 比特，最后通过置换 P 输出.

(a) E 的选位表（表 1.8.3）给出如何将 32 比特转换为 48 比特，总共 8 行 6 列，即若：

$$R_{i-1} = r_1r_2\cdots r_{32}$$

则 $E(R_{i-1}) = r_{32}r_1r_2r_3r_4r_5r_4\cdots r_{32}r_1$

(b) P 是对输入的 32 比特进行置换，产生 32 位输出，如表 1.8.4 所示：

表 1.8.3　E 的选位表

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

表 1.8.4　P 的置换

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

即若输入为　$H = h_1 h_2 \cdots h_{32}$

则输出为　$P(H) = h_{16} h_7 h_{20} h_{21} h_{29} h_{12} \cdots h_{11} h_4 h_{25}$

(c) 8 个 S 盒是将 6 比特的输入映射为 4 比特的输出（表 1.8.5）：

表 1.8.5

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	15	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	S_7
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

以 S_1 盒为例说明它们的功能如下：

若输入为　$b_1b_2b_3b_4b_5b_6$

其中 b_1b_6 两位二进制数表达了 0～3 之间的数，$b_2b_3b_4b_5$ 为四位二进制数表达 0～15 之间的某个数．在 S_1 表中的 b_1b_6 行 $b_2b_3b_4b_5$

列找到一数 m，$0\leqslant m\leqslant 15$，若 $m=m_1m_2m_3m_4$，则 $m_1m_2m_3m_4$ 便是它的 4 比特输出.

例如，输入为 001111，$b_1b_6=01=1$，$b_2b_3b_4b_5=0111=7$，即在 S_1 盒中的第 1 行第 7 列求得数 1，所以 4 比特输出为 0001.

又如对于 S_2 盒输入为 101011，则 S_2 盒中 3 行 5 列元素为 15，故输出为 1111.

(d) 下面介绍如何从 56 比特的密钥 k 计算 k_i，$i=1,2,\cdots,16$. 将 k 分成 8 组，每组 7 比特，另加奇偶校验位，得 64 比特的密钥，即其中第 8，16，24，32，40，48，56，64 位是校验位. k_i 的生成过程如图 1.8.3.

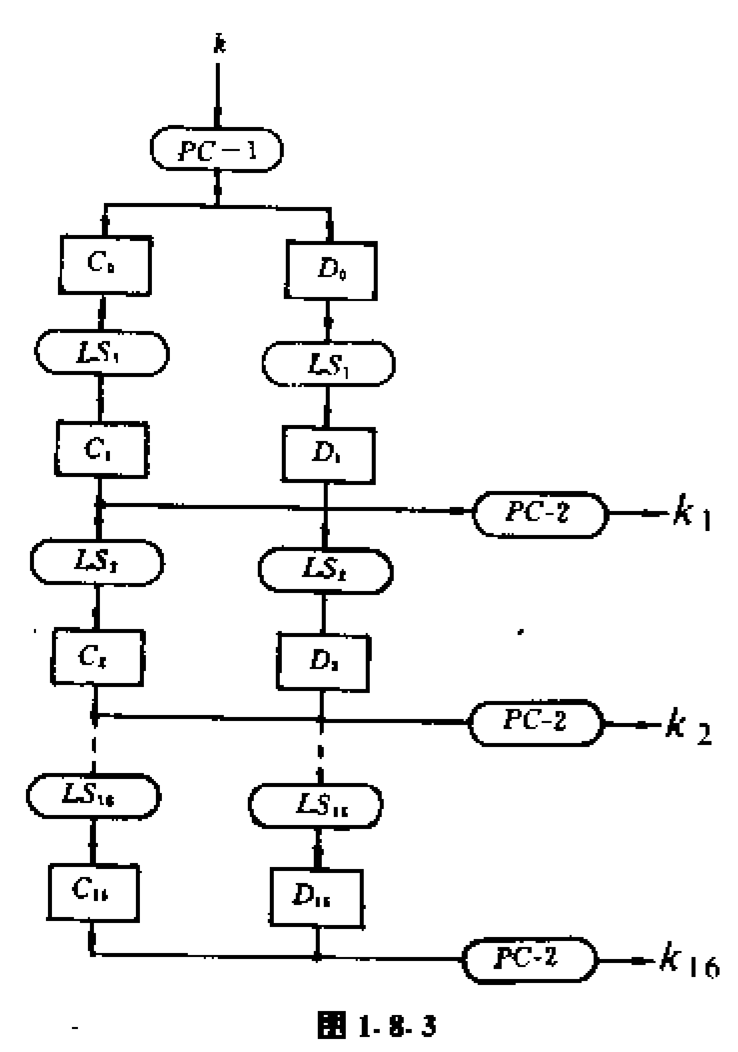


图 1.8.3

其中 PC－1 见表 1.8.6.

表 1.8.6 PC－1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC－1 表为 8 行 7 列，分两部分，前 4 行为 C_0，后 4 行为 D_0，即若密钥：

$$k = k_1k_2\cdots k_{64}$$

则 $$C_0 = k_{57}k_{49}\cdots k_{44}k_{36}$$

$$D_0 = k_{63}k_{55}\cdots k_{12}k_4$$

下面介绍如何从 C_i、D_i 求 C_{i+1}，D_{i+1}，$i=0, 1, 2, \cdots, 15$，设：

$$C_i = c_1c_2\cdots c_{28}, \ D_i = d_1d_2\cdots d_{28}$$

首先要作左移(LS)运算，左移的位数见表 1.8.7.

表 1.8.7

迭代次数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
左移次数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

例如，设

$$C_1 = c_1c_2\cdots c_{28}, \ D_1 = d_1d_2\cdots d_{28}$$

则 $$C_2 = c_2c_3\cdots c_{28}c_1, \ D_2 = d_2d_3\cdots d_{28}d_1$$

C_4 和 D_4 是由 C_3、D_3 左移 2 位而得到，而 C_3、D_3 由 C_2 和

D_2 左移一位而得，故

$$C_4 = c_5c_6\cdots c_{28}c_1c_2c_3c_4,\ D_4 = d_5d_6\cdots d_{28}d_1d_2d_3d_4$$

PC－2 置换如表 1.8.8，共 8 行 6 列 48 个元素．

表 1.8.8　PC－2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

例如，$C_iD_i = b_1b_2\cdots b_{56}$，则

$$k_i = b_{14}b_{17}b_{11}b_{24}\cdots b_{36}b_{29}b_{32}$$

§9　DES 讨论继续

DES 的解密过程和加密过程相似，只不过将密钥的顺序倒过来，即用 DES^{-1} 表 DES 的解密过程，和公式 1.8.1 比较有：

$$DES^{-1} = IP^{-1} \circ T_1 \circ T_2 \circ \cdots \circ T_{16} \circ IP \qquad (1.9.1)$$

则　$DES^{-1} \circ DES = IP^{-1} \circ T_1 \circ T_2 \circ \cdots \circ T_{16} \circ (IP \circ IP^{-1}) \circ T_{16} \circ T_{15} \circ \cdots \circ T_1 \circ IP$

由于　$IP \circ IP^{-1} = I$，I 为恒等置换．

$\therefore$　$DES^{-1} \circ DES = IP^{-1} \circ T_1 \circ T_2 \circ \cdots \circ (T_{16} \circ T_{16}) \circ T_{15} \circ \cdots \circ T_1 \circ IP$

DES(m) 迭代至第 16 次时所得的 64 比特分左右两部分：

$\boxed{L_{15} \oplus f(R_{15}, k_{16})}$ 和 $\boxed{\quad R_{15} \quad}$

若对它继续进行 T_{16}运算，可得

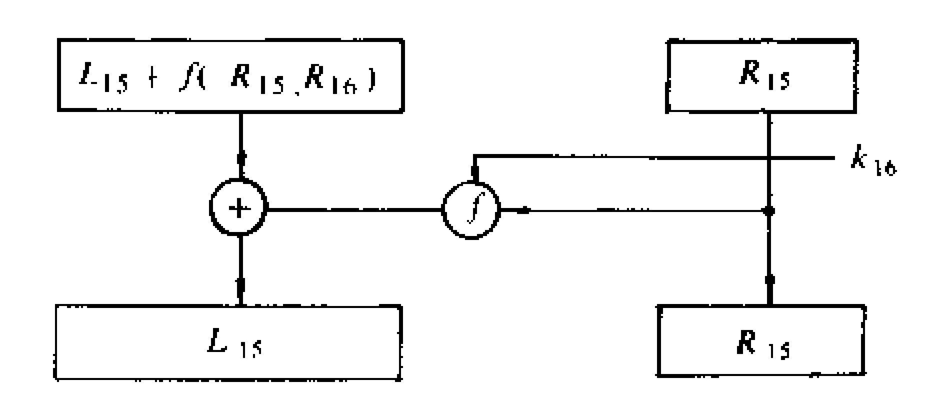


图 1.9.1

左 32 比特为 L_{15}的原因是

$$L_{15} \oplus (f(R_{15}, k_{16}) + f(R_{15}, k_{16})) = L_{15}$$

$$\therefore \quad \text{DES}^{-1} \circ \text{DES}$$

$$= \text{IP}^{-1} \circ T_1 \circ T_2 \circ \cdots \circ T_{15} \circ T_{15} \circ \cdots \circ T_1 \circ \text{IP}$$

同理，DES 迭代至第 15 次时所得的 64 比特分为左右两部分，各 32 比特，为 R_{15} 和 $L_{15} = L_{14} \oplus f(R_{14}, k_{15})$．若对它进行 T_{15}运算得 $L_{15} = R_{14}, R_{15} = L_{14} \oplus f(R_{14}, k_{15}) \oplus f(R_{14}, k_{15}) = L_{14}$. 依此类推，可以证明：

$$\text{DES}^{-1} \circ \text{DES} = \text{IP}^{-1} \circ \text{IP} = \text{I}$$

总而言之，在 DES 中 $f(R_i, k_{i+1})$ 的作用最令人眼花缭乱了. 从 L_i、R_i 到 L_{i+1}、R_{i+1}，有 $L_{i+1} = R_i$，$R_{i+1} = L_i \oplus f(R_i, k_{i+1})$. 算法本身具备了从 L_{i+1}恢复到 L_i、R_i 的可能，因为

$$R_{i+1} \oplus f(R_i, k_{i+1}) = L_i \oplus f(R_i, k_{i+1}) \oplus f(R_i, k_{i+1})$$
$$= L_i$$

而 $\qquad L_{i+1} = R_i.$

特别是 S 盒的设计原理至今仍讳莫如深. 在后面读者将看到若 S 盒的输入改变 1 位, 输出将至少改变 2 位.

例 设明文为 TSINGHUA (清华), 密钥为 BEIJING (北京),将它们化为 ASCII 码分别为 64 位和 56 位的 0—1 字符串:

m=01010100 01010011 01001001 01001110
01000111 01001000 01010101 01000001

k=01000010 01000101 01001001 01001010
01001001 01001110 01000111

密钥 k 补上奇偶校验位 k_8, k_{16}, k_{24}, k_{32}, k_{40}, k_{48}, k_{56}, k_{64}, 得 64 位密钥字符串如下:

01000011 00100011 01010010 00101001
10100100 01001010 00111000 10001110

密钥经 PC—1 得 56 位分为 C_0、D_0 如下:

C_0=10010000 00100101 01011010 0100
D_0=10100111 10010000 11101000 0100

分别左移 1 位得:

C_1=00100000 01001010 10110100 1001
D_1=01001111 00100001 11010000 1001

由 PC—2 产生 k_1 如下:

k_1=01000011 10010100 00000110
00000010 01111011 11100100

明文 m 通过 IP 得:

L_0=11111111 01000011 01011001 11010110
R_0=00000000 00000000 00101100 00011010

$E(R_0)$ = 00000000 00000000 00000000
00010101 10000000 11110100

$E(R_0) \oplus k_1$ = 01000011 10010100 00000110

00010111 11111011 00010000

分成 8 组分别通过 S_1，S_2，…，S_8 盒并输出

S_1	S_2	S_3	S_4
010000	111001	010000	000110
⇓	⇓	⇓	⇓
0011	0000	0001	0011
S_5	S_6	S_7	S_8
000101	111111	101100	010000
⇓	⇓	⇓	⇓
0010	1101	0111	1010

即 S 盒的输出为 32 位字符串.

00110000 00010011 00101101 01111010

经过 P 得 $f(R_0, k_1)$：

10011110 01010010 00100110 10001010

出由此可得第一次迭代后的结果：

L_1 = 00000000 00000000 00101100 00011010

R_1 = 01100001 00010001 01111111 01011100

k_2～k_{16}仅列表如下：

k_2 = 10010000 00100110 00101001 01001100
01101000 10000011

k_3 = 11000001 00010110 00110000 10100110
01100000 01011101

k_4 = 11000100 10011010 11000100 10100011
10010011 11000010

k_5 = 00010010 11110010 01000010 10010100
10000111 00100011

k_6 = 00101000 01010101 01100010 01011110

00001110　01000100

k_7=11100000　01000001　01001101　01011000

11100001　11011000

k_8=00000001　11000011　00010001　00100001

11110100　00001001

k_9=11111110　01000000　01001000　00101010

00011110　11011001

k_{10}=00001010　11000011　00001000　00011011

11010001　00110011

k_{11}=00001000　00011001　00011111　00000111

01001101　00100000

k_{12}=00100101　00101000　01001001　11001000

00101001　01010100

k_{13}=00000011　01101100　10100000　11100001

11000010　10011100

k_{14}=11011000　00101101　10010000　01010001

00010110　10001011

k_{15}=01010100　10100110　00001001　10011110

00010000　00101101

k_{16}=10101000　00101001　10100100　00111001

01000100　10100110

第 2 到第 16 次迭代的结果如下：

L_2=01100001　00010001　01111111　01011100

R_2=11000000　11011000　11010101　01101100

L_3=11000000　11011000　11010101　01101100

R_3=11001110　10101100　10101011　10010010

L_4=11001110　10101100　10101011　10010010

R_4=10100110 00111111 01101001 00001010

L_5=10100110 00111111 01101001 00001010

R_5=11100000 10000001 10000101 01111000

L_6=11100000 10000001 10000101 01111000

R_6=00011100 11100011 11100101 10001010

L_7=00011100 11100011 11100101 10001010

R_7=10001001 11001110 10011100 01100001

L_8=10001001 11001110 10011100 01100001

R_8=11100101 11111000 01101000 11110100

L_9=11100101 11111000 01101000 11110100

R_9=01001110 01111011 00101100 01101100

L_{10}=01001110 01111011 00101100 01101100

R_{10}=11101000 11001011 00111010 11100000

L_{11}=11101000 11001011 00111010 11100000

R_{11}=01111111 10000001 01111100 11101001

L_{12}=01111111 10000001 01111100 11101001

R_{12}=01101001 00001011 00101010 10010100

L_{13}=01101001 00001011 00101010 10010100

R_{13}=00110100 01110011 10001101 10110011

L_{14}=00110100 01110011 10001101 10110011

R_{14}=11100110 01111110 00101100 11000110

L_{15}=11100110 01111110 00101100 11000110

R_{15}=11110110 11101000 00110110 00000111

L_{16}=11110110 11101000 00110110 00000111

R_{16}=00000010 00101011 10001110 01100111

最后得密文 c：

00010011 11011111 10001111 00110100

10001000　10111001　10100001　10100100

§10　码间相关性及其它

(1)　我们感兴趣的是明文的每一位和密文输出的关系，现在假定明文 m 有一位的改变，设 §9 例子中的第 1 位从 0 改为 1，即 m 为

$\dot{1}$1010100　01010011　01001001　01001110

01000111　01001000　01010101　01000001

顶上打 * 号的为改变的位．特别要观察这一位变化是如何传播的．显然 L_0 没有变化，但 R_0 改变为：

0000000$\dot{1}$　00000000　00101100　00011010

$E(R_0)$ 变为

00000000　00$\dot{1}$0$\dot{1}$000　00000000

00010101　10000000　11110100

$E(R_0)\oplus k_1$ 为

S_1	S_2	S_3	S_4
010000	1110$\dot{1}$1	$\dot{1}$10000	000110
⇓	⇓	⇓	⇓
0011	0$\dot{1}$0$\dot{1}$	$\dot{1}\dot{1}\dot{1}$1	0011
S_5	S_6	S_7	S_8
000101	111111	101100	010000
⇓	⇓	⇓	⇓
0010	1101	0111	1010

可见通过 S 盒后有 4 位发生了变化，第 1 次迭代的结果是：

L_1＝00000001　00000000　00101100　00011010

R_1＝01100001　00010001　00111110　01001000

16 次迭代后的密文是：

00100001　11100001　00111010　00100010

10101000　01001000　01000011　10011000

可见明文改变 1 位，导致密文改变 30 位.

下面研究若密钥改变 1 位，观察它对密文的影响. 假定 k_1 从 0 改为 1，即 § 9 中的密钥 k 改为：

11000010　01000101　01001001　01001010

01001001　01001110　01000111

k_1＝01000011　10010100　00010110　00000010

01111011　11100100

$E(R_0)+k_1$ 为：

S_1	S_2	S_3	S_4
010000	111001	010000	010110
⇓	⇓	⇓	⇓
0011	0000	0001	0101
S_5	S_6	S_7	S_8
000101	111111	101100	010000
⇓	⇓	⇓	⇓
0010	1101	0111	1010

第一次迭代的结果是：

L_1＝00000000　00000000　00101100　00011010

R_1＝01100001　01010001　01101111　01011100

最后的密文是：

00000011　00000111　11101010　00010010

10111110　01111100　00100011　01100000

(2) DES 的密钥长度为 56 比特，故有 $2^{56} \approx 7.2 \times 10^{16}$ 个可能密钥. 若对 DES 采取穷举搜索攻击，以每秒搜索 100 万个密钥计，则需要 7.2×10^{10} 秒，等于 2×10^{7} 小时，8.33×10^{5} 天，2282 年.

DES 的唯一解码量不难计算，由于密钥长度为 56 比特，所以 $H(k) = 56$，n 位有意义的报文数目为 $2^{4.19n}$. 若报文由 n 个文字组成，ASCII 码中一个文字用 8 比特来表达，故字符数为 n 的报文数目为 2^{8n}.

$$\therefore \quad H(c) = 8n, \; H(M) = 4.19n$$

$$H(k) = 56 = (8 - 4.19)n$$

$$n = 56/3.81 = 14.7 (\text{字符})$$

从理论上分析只要有 15 个密文字符就足以解出密钥. 15 个字符 120 个比特，大致两个密文块，但唯一解码量没有说明，要做到这一点需要花多少时间，计算上是否可能?

狄菲和赫尔曼于 1977 年提出一种对 DES 进行攻击的设想，基本上是利用穷举搜索的办法. 为此必需造一专用的设备，利用当时的高技术估计 1 微秒能搜索 1 个密钥. $2^{56} \approx 7.2 \times 10^{16}$，而每天共有 $24 \times 3600 = 86400$ 秒 $= 8.64 \times 10^{10}$ 微秒. 若要在一天内完成对所有的密钥搜索，大致需 10^{6} 个芯片并行工作，耗资约 2 千万美元，这当然是当时的估计.

总的说来 DES 的 56 位密钥是不够长的，不足以抗击对它进行的强行攻击.

(3) 弱和半弱密钥.

许多密码都有坏密钥，DES 也不例外. 一是弱密钥，即 $DES_k(m)=DES_k^{-1}(m)$. 例如下面 4 个密钥便是弱密钥：

01	01	01	01	01	01	01	01
1F	1F	1F	1F	1F	1F	1F	1F
E0	E0	E0	E0	E0	E0	E0	E0
FE	FE	FE	FE	FE	FE	FE	FE

其中：01：00000001， 1F：00011111

E0：11100001， FE：11111110

理由是：PC－1 使得 C_0、D_0 或为 00…0，或为 11…1，结果 $k_1=k_2=\cdots=k_{16}$. 故 $DES_k(m)=DES_k^{-1}(m)$. 若 k 是弱密钥，则有：

$$DES_k(DES_k(m))=m$$

半弱密钥指的是存在 k 和 k'，使得：

$$DES_k(m)=DES_{k'}^{-1}(m)$$

即 $$DES_k(DES_{k'}(m))=m$$

k 和 k' 成对构成半弱密钥. C_0 为 1010…10. 而 D_0 或为 00…0，或为 111…11，或为 0101…01，或为 1010…10 的密钥，分别和 C_0 为 0101…01，且 D_0 为 00…00，或为 111…11，或为 0101…01，或为 1010…10 的密钥对偶，这样的半弱密钥对有如下 12 个：

01	FE	01	FE	01	FE	01	FE
FE	01	FE	01	FE	01	FE	01
1F	E0	1F	E0	0E	F1	0E	F1
E0	1F	E0	1F	F1	0E	F1	0E
01	E0	01	E0	01	F1	01	F1
E0	01	E0	01	F1	01	F1	01

$$\left\{\begin{array}{cccccccc} 1F & FE & 1F & FE & 0E & FE & 0E & FE \\ FE & 1F & FE & 1F & FE & 0E & FE & 0E \end{array}\right.$$

$$\left\{\begin{array}{cccccccc} 01 & 1F & 01 & 1F & 01 & 0E & 01 & 0E \\ 1F & 01 & 1F & 01 & 0E & 01 & 0E & 01 \end{array}\right.$$

$$\left\{\begin{array}{cccccccc} E0 & FE & E0 & FE & F1 & FE & F1 & FE \\ FE & E0 & FE & E0 & FE & F1 & FE & F1 \end{array}\right.$$

其中　01：00000001　FE：11111110　$1F$：00011111

$E0$：11100000　$F1$：11110001　$0E$：00001110

第二章　近代密码学研究

§1　问题的提出

由于计算机网络被广泛地应用，通信双方使用的密钥是私下约定的，故一个有 n 个用户的网络，两两间的秘密通信共需 $\frac{1}{2}n(n-1)$ 个密钥，$n=1000$ 时，则 $\frac{1}{2}n(n-1)=499500$，约需要近 50 万个密钥．若考虑到必要的密钥更换，这将是不胜其繁的．更麻烦的还在于每一用户要和其他 999 个用户通信，为了记住这么多的密钥，只好把密钥记在本子上或其它什么载体上，这事情本身就带来了极大的不安全．为此，1976 年狄菲和赫尔曼在著名的“密码学的新方向”一文中提出“公钥密码体系”的设想．在这以前通信双方用的加密密钥和解密密钥都是相同的，所以，也叫做对称密码体制．狄菲和赫尔曼设想假定用户 A 的密钥 $k=[k(a),k(\bar{a})]$，其中 $k(a)$ 为加密密钥，$k(\bar{a})$

是解密密钥，而且，$k(a)$ 和 $k(\bar{a})$ 是不一样的. 此外将加密密钥 $k(a)$ 公开不至于危及解密密钥 $k(\bar{a})$ 的安全，这样可将各用户的加密密钥用公开密钥文件形式向全体用户公开.

若用户 B 欲向 A 送去明文 m，则可查阅公开密码文件得 $k(a)$，利用 $k(a)$ 加密得密文

$$c = E_{k(a)}(m)$$

将 c 送给 A，A 收到 c 后利用只有他掌握的解密密钥 $k(\bar{a})$ 恢复明文 m，即：

$$m = D_{k(\bar{a})}(c)$$

公钥密码通信双方用的加密密钥不同于解密密钥，所以叫做非对称密码.

§2 RSA 公钥密码系统

在狄非和赫尔曼提出公钥的设想后两年，先后由默科(Merkle)和赫尔曼提出了以后称之为 MH-背包公钥密码，和列维斯特(Rivest)，沙米尔(Shamir)和艾德曼(Adleman)联合提出的简称为 RSA 公钥密码系统. 背包公钥密码将在 §4 中介绍. 现在先讨论 RSA 公钥密码. RSA 虽稍后于 MH-背包公钥系统，但它到目前为止仍不失为有希望的一种. RSA 的基础是数论的欧拉定理，它的安全性依赖于分解大数的困难.

(1) 欧拉定理.

定理 2.2.1 若整数 a 和 m 互素，则

$$a^{\varphi(m)} \equiv 1 \pmod{m}$$

其中 $\varphi(m)$ 是比 m 小但与 m 互素的正整数的个数.

证：设 $\varphi(m) = k$. 又设 r_1，r_2，…，r_k 是小于 m 并与 m 互

素的数，且由于 a 是与 m 互素的整数，则 ar_1，ar_2，…，ar_k 也和 m 互素且两两不相同. 若

$$ar_i \equiv ar_j (\mathrm{mod}\ m)$$

则根据数论定理，因 a 和 m 互素，所以存在 $\bar{a}$ 满足

$$\bar{a}a \equiv 1(\mathrm{mod}\ m)$$

$$\therefore \quad \bar{a}ar_i \equiv \bar{a}ar_j (\mathrm{mod}\ m)$$

即 $\quad r_i \equiv r_j (\mathrm{mod}\ m)$

与假定矛盾，所以

$$a^k r_1 r_2 \cdots r_k \equiv r_1 r_2 \cdots r_k (\mathrm{mod}\ m)$$

但 $r_1 r_2 \cdots r_k$ 和 m 互素，故

$$a^k \equiv 1(\mathrm{mod}\ m)$$

(2) RSA 算法.

1. 取两个素数 p 和 q(保密)，
2. 计算 $n=pq$ (公开)，$\varphi(n)=(p-1)(q-1)$ (保密).
3. 随机选取整数 e，满足 $\gcd(e,\varphi(n))=1$ (公开).
4. 计算 d，满足 $de \equiv 1(\mathrm{mod}\ \varphi(n))$(保密).

利用 RSA 加密第一步需将明文数字化，并取长度小于 $\log_2 n$ 位的数字作明文块.

加密算法 $\quad c = E(m) \equiv m^e (\mathrm{mod}\ n)$

解密算法 $\quad D(c) \equiv c^d (\mathrm{mod}\ n)$

下面证明解密过程是正确的.

证：对于任何 k 及任何 m ($<n$)，恒有

$$m^{k\varphi(n)+1} \equiv m(\mathrm{mod}\ n)$$

若 $(m,n)=1$，则由欧拉定理可知：

$$m^{\varphi(n)} \equiv 1(\mathrm{mod}\ n)$$

若 $(m,n)>1$，由于 $n=pq$，故 (m,n) 必含 p、q 之一. 设 $(m,n)=p$，或 $m=cp$，$1 \leqslant c < q$，由欧拉定理：

$$m^{\varphi(q)} \equiv 1 (\mathrm{mod}\ q)$$

因此对任何 k，总有

$$m^{k(q-1)} \equiv 1 (\mathrm{mod}\ q)$$

$$m^{k(p-1)(q-1)} \equiv (1)^{k(p-1)} \equiv 1 (\mathrm{mod}\ q)$$

即 $$m^{k\varphi(n)} \equiv 1 (\mathrm{mod}\ q)$$

或 $$1 = m^{k\varphi(n)} + hq$$

其中 h 是某个整数. 由假定 $m=cp$，

$$\therefore \quad m = m^{k\varphi(n)+1} + hcpq$$

这就证明了

$$m = m^{k\varphi(n)+1} (\mathrm{mod}\ n)$$

举例如下：

设 $p = 43$， $q = 59$， $n = pq = 43 \times 59 = 2537$

$$\varphi(n) = 42 \times 58 = 2436, \quad \text{取 } e = 13$$

解方程 $de \equiv 1 (\mathrm{mod}\ 2436)$

$$2436 = 13 \times 187 + 5, \quad 13 = 2 \times 5 + 3$$

$$5 = 3 + 2, \quad 3 = 2 + 1$$

$$\therefore \quad 1 = 3 - 2, \quad 2 = 5 - 3, \quad 3 = 13 - 2 \times 5$$

$$5 = 2436 - 13 \times 187$$

$$\therefore \quad 1 = 3 - 2 = 3 - (5 - 3) = 2 \times 3 - 5$$

$$= 2(13 - 2 \times 5) - 5$$

$$= 2 \times 13 - 5 \times 5 = 2 \times 13 - 5(2436 - 13 \times 187)$$

$$= 937 \times 13 - 5 \times 2436$$

即 $$937 \times 13 \equiv 1 (\mathrm{mod}\ 2436)$$

取 $e=13$ 时，$d=937$

若有明文：public key encryptions.

先将明文分块为：

pu bl ic ke ye nc ry pt io ns

将明文数字化得：

1520　0111　0802　1004　2404

1302　1724　1519　0814　1418

利用加密得密文：

0095　1648　1410　1299　1365

1379　2333　2132　1751　1289

(3) 若 $n=pq$ 被因子分解，则 RSA 便被击破. 因为，如果 p、q 已知，则 $\varphi(n)=(p-1)(q-1)$ 便可算出. 解密密钥 d 关于 e 满足：

$$de \equiv 1(\mathrm{mod}\ \varphi(n))$$

故 d 便不难求得. 因此 RSA 的安全依赖于因子分解的困难性. 目前因子分解速度最快的方法，其时间复杂性为：

$$\exp(\mathrm{sqrt}(\ln(n)\ln\ln(n)))$$

列维斯特、沙米尔和艾德曼建议取 p 和 q 为 100 位 10 进制数，这样 n 为 200 位 10 进制数. 要分解 200 位的 10 进制数按每秒 10^7 次运算的超高速电子计算机，也要 10^8 年. 近来对大数分解算法的研究引起了重视. 1990 年的最新结果有 150 位的特殊类型的数（第 9 个费尔玛(Fermat)数）已被成功分解.

若 n 被分解成功，则 RSA 便被攻破. 但还不能证明对 RSA 攻击的难度和分解 n 相当，故对 RSA 的攻击的困难程度不比大数分解更大. 当然，若从求 $\varphi(n)$ 入手对 RSA 进行攻击，它的难度和分解 n 相当. 已知 n，求得 $\varphi(n)$，则 p 和 q 可以求得. 因为

$$\begin{cases} n-\varphi(n)+1=p+q \\ \sqrt{(p+q)^2-4n}=p-q \end{cases}$$

为了安全起见，对 p 和 q 还要求：

(a) p 和 q 的长度相差不大.

(b) $p-1$ 和 $q-1$ 有大素数因子.

(c) $(p-1, q-1)$ 很小.

满足这些条件的素数称做安全素数.

(4) 下面给出一个值得引起注意的例子.

令 $m_k = \mathrm{RSA}(m_{k-1}) = m_{k-1}^e(\mathrm{mod}\,n)$

$p=17$, $q=11$, $n=pq=17\times 11=187$, $e=7$, $m=123$

可以证明 m 经过 RSA 继续变换可得 $m_4=m$，即，连续 4 次进行 RSA 变换恢复到原文.

$m_1 = 123$, $123^7 \equiv 183(\mathrm{mod}\ 187)$

$m_2 = 183$, $183^7 \equiv 72(\mathrm{mod}\ 187)$

$m_3 = 72$, $72^7 \equiv 30(\mathrm{mod}\ 187)$

$m_4 = 30$, $30^7 \equiv 123(\mathrm{mod}\ 187)$

(5) RSA 加、解密变换都要进行 mod n 的幂运算.

求 $x^r(\mathrm{mod}\ p)$ 可通过对指数 r 的 2 进制数化来实现. 例如求 $11^7(\mathrm{mod}\ 17)$, $7=(111)_2$, 即

$$7 = 2^2 + 2^1 + 2^0 = 2^2 + 2 + 1$$

故

$$11^7(\mathrm{mod}\ 17) = (11)^{2^2} \cdot 11^2 \cdot 11(\mathrm{mod}\ 17)$$

下面给出一种比较实用的方法. 在给出方法之前，先通过实例观察幂运算的计算步骤.

$$p = 1520^{13}(\mathrm{mod}\ 2537) = (1520)^{2\times 6+1}(\mathrm{mod}\ 2537)$$
$$= (1520^2)^6 1520(\mathrm{mod}\ 2537)$$

但 $(1520)^2 \equiv 1730(\mathrm{mod}\ 2537)$

$\therefore$ $p \equiv (1730)^6 1520(\mathrm{mod}\ 2537)$

$= (1730^2)^3 1520(\mathrm{mod}\ 2537)$

但 $1730^2 \equiv 1777(\mathrm{mod}\ 2537)$

$\therefore$ $p \equiv (1777)^3 \cdot 1520(\mathrm{mod}\ 2537)$

$$= 1777^2 \cdot (1777 \cdot 1520) (\text{mod } 2537)$$

但 $1777^2 (\text{mod } 2537) \equiv 1701$

$1777 \cdot 1520 (\text{mod } 2537) \equiv 1672$

$\therefore \quad p \equiv 1701 \cdot 1672 (\text{mod } 2537) \equiv 95$

现在模拟这个过程给出计算 $x^r (\text{mod } p)$ 流程如下：

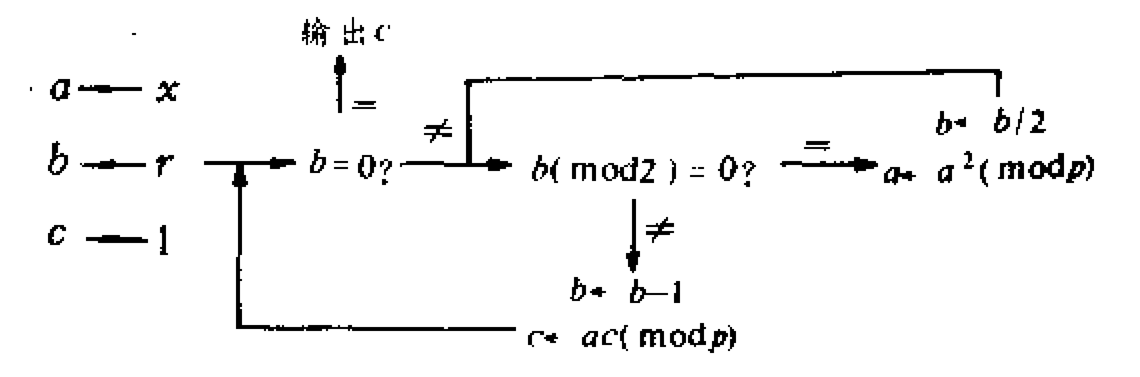


图 2.2.1

例 求 $(16)^{15} (\text{mod } 4731)$

(1) $a \leftarrow 16$，$b \leftarrow 15$，$c \leftarrow 1$

(2) $b \leftarrow 14$，$c \leftarrow 16$

(3) $b \leftarrow 7$，$(16)^2 (\text{mod } 4731) \equiv 256$， $a \leftarrow 256$

(4) $b \leftarrow 6$，$256 \times 16 \ (\text{mod } 4731) = 4096$，$c \leftarrow 4096$

(5) $b \leftarrow 3$，$(256)^2 (\text{mod } 4731) = 4033$， $a \leftarrow 4033$

(6) $b \leftarrow 2$，$4033 \times 4096 \ (\text{mod } 4731) \equiv 3247$， $c \leftarrow 3247$

(7) $b \leftarrow 1$，$4033^2 (\text{mod } 4731) \equiv 4642$， $a \leftarrow 4642$

(8) $b \leftarrow 0$，$4642 \times 3247 \ (\text{mod } 4731) \equiv 4339$， $c \leftarrow 4339$

(9) $\because \quad b = 0 \quad \therefore \quad 16^{15} (\text{mod } 4731) \equiv 4339$

§3 勒宾 (Rabin) 密码系统

前一节已讨论了若分解 n，则 RSA 便被攻破，即 RSA 的破

译的难度不超过大数分解. 勒宾提出对 RSA 的一种修正，可以证明对它的破译等价于对 n 的分解. RSA 是选择加密密钥 e 满足 $1 \leqslant e < \varphi(n)$ 且 $(e, \varphi(n)) = 1$. 勒宾是取 $e=2$，即加密算法是

$$c \equiv m^2 (\text{mod } n)$$

由于 $n = pq$，故

$$c \equiv m^2 (\text{mod } p), \quad c \equiv m^2 (\text{mod } q)$$

令 QR_n 表示模 n 的平方剩余集合，即

$$QR_n \triangleq \{a \mid \exists\ x \in Z, x^2 \equiv a (\text{mod } n)\}$$

$\mathbf{Z}$ 表示整数集合，即若存在 $x \in \mathbf{Z}$ 满足

$$x^2 \equiv a (\text{mod } n)$$

则称 a 属于 QR_n，或写成 $a \in QR_n$，表 a 为模 n 的平方剩余，否则 $a \notin QR_n$，或称 a 为非平方剩余.

引理 2.3.1 若 p 为奇素数，$p \nmid a$，则

$$x^2 \equiv a (\text{mod } p)$$

或无解或有两个模 p 不同余的解.

证：若 $x^2 \equiv a$ (mod p) 有解 x_1 $(0 < x_1 < p)$，不难验证 $-x_1$ 也满足这个同余方程，而且 x_1 和 $-x_1$ 不是模 p 同余，如若不然，$x_1 \equiv -x_1 (\text{mod } p)$ 则

$$2x_1 \equiv 0 (\text{mod } p)$$

这跟 $(0 < x_1 < p)$，且 p 是奇素数的假定相矛盾.

下面证只有这两个，别无其它解. 若

$$x_2^2 \equiv a (\text{mod } p)$$

则 $$x_1^2 \equiv x_2^2 (\text{mod } p)$$

$$\therefore \quad x_1^2 - x_2^2 \equiv (x_1 - x_2)(x_1 + x_2) \equiv 0 (\text{mod } p)$$

$$\therefore \quad p \mid (x_1 - x_2) \quad 或 \quad p \mid (x_1 + x_2)$$

这就证明了

$$x_1 \equiv x_2 (\bmod p) \quad 或 \quad x_1 \equiv -x_2 (\bmod p)$$

定理 2.3.1 若 p 是奇素数，则整数 1，2，…，$p-1$ 中正好有（$p-1$）/2 个属于 QR_p，其余（$p-1$）/2 个属于非平方剩余.

证：计算 1，2，…，$p-1$ 的平方模 p 的最小正剩余. 因为只有 $p-1$ 个非零的剩余，且 $x^2 \equiv a (\bmod p)$ 的解的数目或为零或有不同余的两个. 故正好有（$p-1$）/2 个模 p 的平方剩余. 即存在（$p-1$）/2 个 a，满足 $1 \leqslant a \leqslant p-1$，使得 $a \in QR_p$. 其余的（$p-1$）/2 个为非平方剩余.

例 $p=5$，$1^2 \equiv 1$（mod 5），$2^2 \equiv 4$（mod 5），$3^2 \equiv 4$（mod 5），$4^2 \equiv 1$（mod 5），所以 1 和 4 属于 QR_5，2 和 3 则属于非平方剩余.

定理 2.3.2 （中国剩余定理） 设 m_1，m_2，…，m_k 是两两互素的正整数，则同余方程组

$$x \equiv b_i (\bmod m_i) \qquad i = 1, 2, \cdots, k$$

模 $m_1 \cdot m_2 \cdot \cdots \cdot m_k$ 有唯一解.

证：证明是构造性的，即证明的过程也是给出解的过程. 令：

$$M = m_1 m_2 \cdots m_k$$

$$M_i = M/m_i = m_1 m_2 \cdots m_{i-1} m_{i+1} \cdots m_k, \quad i = 1, 2, \cdots, k$$

求 y_j 使之满足：

$$M_j y_j \equiv 1 (\bmod m_j), \quad j = 1, 2, \cdots, k$$

由于 M_j 和 m_j 互素，所以 y_j 是存在的. 可以证明

$$x = b_1 M_1 y_1 + b_2 M_2 y_2 + \cdots + b_k M_k y_k$$

满足同余方程组

$$x \equiv b_i (\bmod m_i), \quad i = 1, 2, \cdots, k$$

证明过程请注意：只要 $j \neq h$，则 m_j 除尽 M_h，所以

$$M_h \equiv 0 (\bmod m_j)$$

故　$x \equiv b_h M_h y_h (\bmod m_h)$

但　$M_h y_h \equiv 1 (\bmod m_h)$

所以得证

$$x \equiv b_h (\bmod m_h), \quad h = 1,2,\cdots,k$$

以下证明解是唯一的．如若不然，有 x_1 和 x_2 都满足：

$$x \equiv b_j (\bmod m_j), \quad j = 1,2,\cdots,k$$

且 $1 \leqslant x_1 \leqslant M$，$1 \leqslant x_2 \leqslant M$，则

$$x_1 - x_2 \equiv 0 (\bmod m_j), \quad j = 1,2,\cdots,k$$

这就证明了：$m_j | (x_1 - x_2)$，　$j = 1,2,\cdots,k$

即　　$M | (x_1 - x_2)$

$$\therefore \quad x_1 \equiv x_2 (\bmod M)$$

至此命题获证．

有了中国剩余定理可知以下重要事实：

已知：

$$c \equiv m^2 (\bmod n)$$

求 m，导至解

$$x^2 \equiv c (\bmod p), \quad x^2 \equiv c (\bmod q)$$

每一个都将有两个解，即：

$$x \equiv m (\bmod p), \quad x \equiv -m (\bmod p)$$

$$x \equiv m (\bmod q), \quad x \equiv -m (\bmod q)$$

根据中国剩余定理：

$$x^2 \equiv c (\bmod n)$$

将有四个解，也就是说勒宾密码的已知密文对应的明文不唯一．

勒宾密码还有一个重要的结论，即对它的攻击的困难程度等价于分解 n．

定理 2.3.3　求解方程：

$$x^2 \equiv a \ (\bmod n) \qquad (2.3.1)$$

与分解 n 是等价的.

证：如果能从 n 分解出 p 和 q，则导致解：

$$\begin{cases} x^2 \equiv a(\mathrm{mod}\ p) \\ x^2 \equiv a(\mathrm{mod}\ q) \end{cases} \tag{2.3.2}$$

在 $p = q = 3(\mathrm{mod}\ 4)$ 时，后者已有有效算法. 反之，若(2)有解 $\pm x_1$，$\pm x_2$，$x_1 \not\equiv x_2$ $(\mathrm{mod}\ n)$，由于

$$x_1^2 \equiv x_2^2(\mathrm{mod}\ n)$$

$$\therefore \quad (x_1 + x_2)(x_1 - x_2) \equiv 0(\mathrm{mod}\ n)$$

注意到 p 和 q 不能同时整除 (x_1+x_2) 和 (x_1-x_2)，故下面两种可能有且仅有一种可能发生，即：

(a) $p|(x_1 + x_2)$，$q|(x_1 - x_2)$

(b) $p|(x_1 - x_2)$，$q|(x_1 + x_2)$

总之，求 n 和 x_1+x_2（或 x_1-x_2）的最大公因子便是 p（或 q）.

§4 数字签名

传统的对称密码有一弱点. 由于加密密钥和解密密钥是一样的，所以，当 A 送 B 以密文 $c = E_k(m)$，B 有能力对密文（实际上也就是对明文）进行篡改. 如果 A 和 B 之间对"A 是否向 B 送去 m"发生争执时，困难也随之而生. 最好有一种密码，不仅具有保密的功能，而且还具有以下的功能：(a) B 不能篡改明文的内容，(b) A 不能否认他向 B 送过 m. RSA 具有这种能力. 假如 A 送给 B 以密文 c：

$$c = E_B(D_A(m))$$

即 A 先用他自己掌握的解密密钥对 m 作变换得 $D_A(m)$，接着

对 $D_A(m)$ 用 B 的加密密钥进行加密，得密文 $c=E_B(D_A(m))$，而 B 的加密密钥可以从公钥文件中查到. B 收到 c 后先用他自己掌握的解密密钥对密文进行解密得：

$$D_B(c)=D_B(E_B(D_A(m)))=D_A(m)$$

再对 $D_A(m)$ 用 A 的加密密钥作变换得：

$$E_A(D_B(c))=E_A(D_A(m))=m$$

从而恢复了明文 m. 如若 A 对是否送去 m 提出异议，则 B 可出示他收到的密文 c. 因为密文 $E_B(D_A(m))$ 中只有 A 才掌握他自己的解密密钥，也就是自己的解密算法 D_A. 正因为如此，B 也不可能篡改明文的内容. 它颇类似于传统的送去的信件签上名一样，故称这样的功能为数字签名. 并不是任何密码系统都具有这样的能力.

§5 背包问题和NP理论

(1) 背包问题.

当狄菲和赫尔曼提出公钥密码系统的设想时，还没有一个这样的实例. 两年后首先由默科和赫尔曼提出一基于组合数学中背包问题的公钥密码系统. 这个背包系统用MH-背包密码系统表示.

所谓背包问题是这样的：已知一长度为 b 的背包，及长度分别为 a_1，a_2，…，a_n 的 n 个物品. 假定这些物品的半径和背包相同，要求从这 n 个物品中选出若干个正好装满这背包. 这问题导致求 $x_i=0$ 或 1，$i=1$，2，…，n，使满足

$$\sum_{i=1}^{n}a_ix_i=b$$

其中 a_1，a_2，…，a_n 和 b 都是正整数.

背包问题是著名的难题，至今还没有好的求解方法. 若对 2^n 种所有可能性进行穷举搜索，实际上是不可能的，以 $n=100$ 为例：

$$2^{100}=1.27\times10^{30}$$

以每秒搜索 10^7 种方案的超高速电子计算机进行穷举，一年只能完成 3.1536×10^{14} 次，所以共要

$$1.27\times10^{30}/(3.1536\times10^{14})=4.02\times10^{15}(\text{年}).$$

算法复杂性理论已经证明：背包问题是属于 NP 完全类，也就是说它是 NP 类问题中难度最大的一类. 这一类问题还没有有效的算法. 对 2^n 种可能穷举搜索不是好算法，因为它不可能在实际允许时间内完成.

必须指出的是并非所有的背包问题都没有有效算法. 如若序列 a_1，a_2，…，a_n 满足条件：

$$a_i>\sum_{j=1}^{i-1}a_j,\quad (i=2,3,\cdots,n)$$

时，有多项式解法. 这样的序列称之为超递增序列，例如：

$$1,2,4,8,\cdots,2^n$$

就是超递增的序列.

$$x_1+2x_2+4x_3+8x_4+16x_5+32x_6=43$$

则因 $43>32$，故 $x_6=1$，以之代入得：

$$x_1+2x_2+4x_3+8x_4+16x_5=11$$

$11<16$，只能有 $x_5=0$

$$x_1+2x_2+4x_3+8x_4=11$$

$$\therefore\quad x_1=x_2=x_4=x_6=1$$

$$x_3=x_5=0$$

是问题的解.

(2) NP 类和 NP 完全类.

在介绍 NP 类问题时先解说什么叫 P 类问题. 设 $G=(V, E)$ 是已知有 n 个顶点的图, 即 $|V|=n$, $A=(a_{ij})_{n\times n}$ 是图 G 的邻接矩阵:

$$a_{ij}=\begin{cases}1, & 若(v_i,v_j)\in E\\ 0, & 否则\end{cases}$$

比如 G 为图 2.5.1.

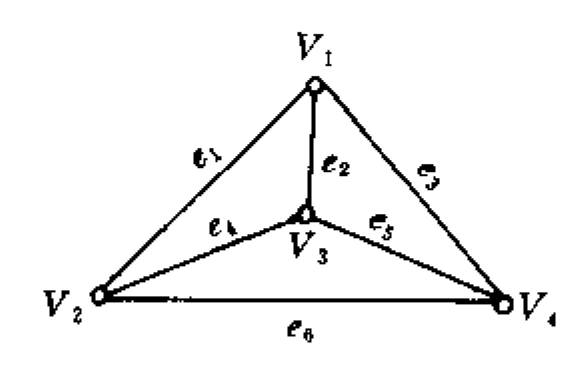


图 2. 5. 1

其中 $V=\{v_1, v_2, v_3, v_4\}$, $E=\{e_1, e_2, e_3, e_4, e_5, e_6\}$, 则邻接矩阵为:

$$A=\begin{array}{c} \\ v_1\\ v_2\\ v_3\\ v_4\end{array}\begin{array}{c}\begin{matrix}v_1 & v_2 & v_3 & v_4\end{matrix}\\ \begin{pmatrix}0&1&1&1\\1&0&1&1\\1&1&0&1\\1&1&1&0\end{pmatrix}\end{array}$$

$$A^2=\begin{pmatrix}0&1&1&1\\1&0&1&1\\1&1&0&1\\1&1&1&0\end{pmatrix}\cdot\begin{pmatrix}0&1&1&1\\1&0&1&1\\1&1&0&1\\1&1&1&0\end{pmatrix}=\begin{pmatrix}3&2&2&2\\2&3&2&2\\2&2&3&2\\2&2&2&3\end{pmatrix}$$

一般有

$$a_{ij}^{(2)} = \sum_{k=1}^{n} a_{ik}a_{kj}$$

所以，$a_{ij}^{(2)}$表示从 v_i 出发，通过中间某一 v_k 点到 v_j 的路径数目，即求：

$$v_i \longrightarrow v_k \longrightarrow v_j$$

的路径数. 求 A^2 需要 n^3 次乘法运算，或许说算法的时间复杂性是 $O(n^3)$.

若其时间复杂性是问题规模 n 的多项式，这样的问题称为属于多项式类，或说属于 P 类. 所以，P 类问题有有效算法.

但是，背包问题直到目前还没有找到多项式的算法. 若采取穷举的强行搜索方法，其复杂性为 $O(2^n)$ ，是典型的指数型问题. 求背包问题的解有困难，若给定一组 0—1 序列 $(x_1, x_2, \cdots, x_n)$，问它是否为背包问题，

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$$

的解，验证可在 n 次加法和 n 次乘法内完成.

问题的解可在规模 n 的多项式时间内验证完毕的问题，称为属于 NP 类.

显然，P 类问题属于 NP 类，即：

$$\mathrm{P} \subseteq \mathrm{NP}$$

但是否 NP＝P？这问题尚未解决. 像背包问题属于 NP，但不能判定它是否属于 P 一样.

(3) NP 完全类及库克(Cook)定理.

库克提出一著名的“可满足性问题”，并证明了任何 NP 问题都可以在多项式时间内转化为求解可满足性问题，这就证明了可满足性问题是 NP 类中难度最大的一类问题，从而奠定了算法复杂性理论的基础.

先通过一个例子引进必要的概念，逻辑表达式：$(\bar{x}_1 \vee \bar{x}_3)$

$\wedge$ ($\overline{x}_1 \vee x_2 \vee x_3$) $\wedge$ ($x_1 \vee x_3 \vee x_4$) $\wedge$ ($x_2 \vee \overline{x}_3$) 中 x_1、x_2、x_3、x_4 为逻辑变量，它只取 T 或 F 两个值.

$\overline{x}_1$、$\overline{x}_2$、$\overline{x}_3$、$\overline{x}_4$ 分别是 x_1、x_2、x_3、x_4 的"非". 其中 $\overline{x}_1 \vee \overline{x}_3$、$\overline{x}_1 \vee x_2 \vee x_3$、$x_1 \vee x_3 \vee x_4$、$x_2 \vee \overline{x}_3$ 称为该逻辑表达式的子句. 子句的特点是由逻辑变量作逻辑和构成的，而且每一子句中不同时出现 x_i 和它的"非"$\overline{x}_i$.

若一逻辑表达式可表为若干子句 $c_1, c_2, \cdots, c_m$ 的逻辑乘，即：

$$f = c_1 \wedge c_2 \wedge \cdots \wedge c_m$$

而每一个子句均为逻辑变量（包括它的非）的逻辑和，则称 f 的这种形式为合取范式.

可满足性问题：设 $V = \{x_1, x_2, \cdots, x_n, \overline{x}_1, \overline{x}_2, \cdots, \overline{x}_n\}$，$c_1$，$c_2$，…，$c_m$ 是 V 的有限子集（即为子句）. 每个 c_i 中不出现 V 中互反的一对（即若 $x \in c_i$，则 $\overline{x} \notin c_i$），$i=1, 2, \cdots, m$. 所谓可满足性问题即确定一集合 $S \subseteq V$，满足：

(a) S 中不包含互反的一对，即 $x \in S$，$\overline{x} \notin S$；

(b) $S \cap c_i \neq \varphi$，$i=1, 2, \cdots, m$

从数理逻辑角度来看，S 中元素若都取 T，由 $S \cap c_i \neq \varphi$ 得 c_1，c_2，…，c_m 都取值 T，所以

$$c_1 \wedge c_2 \wedge \cdots \wedge c_n = T$$

可满足性问题显然属于 NP 类.

下面叙述库克定理.

定理 2.5.1 若 L 是属于 NP 类的问题，则 L 可在多项式时间内转变为与之相对应的某一可满足性问题.

定理的证明甚繁，可见于算法复杂性理论的专著，这里从略. 但定理说明了一个十分深刻的事实，可满足性问题是 NP 类中难度最大的. 以它作为标准，称它属于 NP 完全类，或用 NPC

表之. 只要可满足性问题有多项式解法，则整个NP类都有多项式解法.

NP完全类实际上是NP类中难度最大的.

若有一问题 p_1 属于NP，而且，另有一个属于NP完全的问题 p_2，若 p_2 可在多项式时间内转化为求解 p_1，则 p_1 也属于NPC.

现在已经证明有数目十分可观的一类组合数学问题属于NPC. 它们中只要有一个问题有多项式解法，则NP类全体都有多项式算法，即NP=P. 也就是说整个NPC类问题难度相当. 但没找到这样的算法，有很大可能NP根本不等于P，但逻辑上还没有得到证明. 所以，在数学或计算机科学中"NP=P?"仍是十分引人入胜的重大理论问题. 这就是算法复杂性理论要解决的问题.

算法复杂性理论已成为近代密码学的理论基础. 如若破译一密码系统，等价于解一个NPC问题，那么这个密码系统的安全便有了保障.

§6 MH背包公钥密码系统

背包公钥密码系统是选取一组正整数 $a_1, a_2, \cdots, a_n$ 作为公钥予以公布，$m=m_1m_2\cdots m_n$ 是 n 位0，1明文符号串. 利用公钥加密如下：

$$c = a_1m_1 + a_2m_2 + \cdots + a_nm_n$$

从密文 c 求明文 $m_1, m_2, \cdots, m_n$ 等价于解背包问题，例如已知：

$$a_1=28,\quad a_2=32,\quad a_3=11,\quad a_4=08$$

$$a_5=71,\quad a_6=51,\quad a_7=43,\quad a_8=67$$

明文：

$$m_1=10110110，m_2=11001001$$

分别加密得密文：

$$c_1=28+11+8+51+43=141$$

$$c_2=28+32+71+67=198$$

但是，MH 背包公钥系统选的背包序列 $a_1,a_2,\cdots,a_n$ 是由超递增序列进行以下变换得到的.

设 $b_1, b_2, \cdots, b_n$ 是超递增序列，即：

$$b_i>\sum_{j=1}^{i-1}b_j \qquad i=2,3,\cdots,n$$

选取 $m>2b_n$，w 和 m 互素，$\overline{w}$ 满足 $w\overline{w}\equiv 1 \pmod m$. 作变换（我们称之为默科—赫尔曼变换）：

$$a_k\equiv wb_k \pmod m), \quad k=1,2,\cdots,n \qquad (2.6.1)$$

于是从超递增序列：$b_1, b_2, \cdots, b_n$ 得序列：$a_1, a_2, \cdots, a_n$. 一般说来 $\{a_i\}$ 表面上再不具有超递增的特性，MH 背包公钥密码系统便是以这样的序列$\{a_i\}$作为公钥的.

假定已知：

$$a_1m_1+a_2m_2+\cdots+a_nm_n=c$$

c 为 $m=m_1m_2\cdots m_n$ 的密文，其中 $m_1, m_2, \cdots, m_n$ 是 n 位的 0，1 符号串，则：

$$\begin{aligned}\overline{w}c &= \overline{w}a_1m_1+\overline{w}a_2m_2+\cdots+\overline{w}a_nm_n\\ &\equiv b_1m_1+b_2m_2+\cdots+b_nm_n \pmod m \end{aligned} \qquad (2.6.2)$$

这是超递增的背包问题.

例 1，3，7，13，26，65，119，267 是超递增序列.

$$1+3+7+13+26+65+119+267=501$$

取 $m=523$，$w=467$，$\overline{w}=28$，

$$a_1\equiv 467\times 1\equiv 467 \pmod{523}$$

$$a_2 \equiv 467 \times 3 \equiv 355 \pmod{523}$$
$$a_3 \equiv 467 \times 7 \equiv 131 \pmod{523}$$
$$a_4 \equiv 467 \times 13 \equiv 318 \pmod{523}$$
$$a_5 \equiv 467 \times 26 \equiv 113 \pmod{523}$$
$$a_6 \equiv 467 \times 65 \equiv 21 \pmod{523}$$
$$a_7 \equiv 467 \times 119 \equiv 135 \pmod{523}$$
$$a_8 \equiv 467 \times 267 \equiv 215 \pmod{523}$$

（467，355，131，318，113，21，135，215）作为公钥予以公布．对于明文 $m=10101100$ 加密得密文：

$$467+131+113+21=732$$

接收方收到密文 732 后，乘以 $\overline{w}=28 \pmod{523}$，得

$$732\times28=20496\equiv99 \pmod{523}$$

解超递增序列背包问题：

$$m_1+3m_2+7m_3+13m_4+26m_5+65m_6+119m_7+267m_8\equiv99$$

$$\therefore \quad m_1=m_3=m_5=m_6=1$$

$$m_2=m_4=m_7=m_8=0$$

即得明文　10101100

默科和赫尔曼的背包公钥密码发表后，声称：谁能攻破它奖励 50 美元．他们利用默科—赫尔曼变换虽将超递增序列的特性隐蔽起来，但得到的序列毕竟不是“正宗”的，终究露出了“尾巴”．两年后被沙米尔(Shamir)抓住并将它破译．默科和赫尔曼又企图通过多次的默科—赫尔曼变换试图将它变得彻底些，再次悬赏给破译者．再过两年，新的破译方法解决了低密度的背包公钥密码问题(这将在 § 10 中给予介绍)，使 MH 背包公钥密码系统遭受到致命的打击．

§7 MH 背包公钥的简单变形

前面讲到的背包公钥密码系统可将它推广到 $GF(p^n)$ 域. 设

$$
\begin{aligned}
b_1(x) &= b_{10} + b_{11}x + b_{12}x^2 + \cdots + b_{1k}x^k \\
b_2(x) &= b_{20} + b_{21}x + b_{22}x^2 + \cdots + b_{2k}x^k \\
&\cdots\cdots \\
b_n(x) &= b_{n0} + b_{n1}x + b_{n2}x^2 + \cdots + b_{nk}x^k
\end{aligned}
\tag{2.7.1}
$$

使得矩阵

$$
\boldsymbol{B} = \begin{pmatrix} b_{10} & b_{11} & b_{12} & \cdots & b_{1k} \\ b_{20} & b_{21} & b_{22} & \cdots & b_{2k} \\ b_{30} & b_{31} & b_{32} & \cdots & b_{3k} \\ \vdots & \vdots & \vdots & & \vdots \\ b_{n0} & b_{n1} & b_{n2} & \cdots & b_{nk} \end{pmatrix} = (\boldsymbol{b}_0\boldsymbol{b}_1\cdots\boldsymbol{b}_k) \tag{2.7.2}
$$

的每一列构成一超递增序列. 即 $\boldsymbol{b}_i = (b_{1i}, b_{2i}, b_{ni})$ 是一超递增序列, $i=0, 1, \cdots, k$.

令 $M=\max\limits_{i,j}\{b_{ij}\}$

p 为大于 $2M$ 的素数. $m(x)$ 是 $GF(p)$ 域上的 $k+1$ 次不可化约的多项式. $w(x)$ 是方次不超过 k 的多项式. 在 $GF(p)$ 域上 mod $m(x)$ 构成 $GF(p^{k+1})$ 域. 在 $GF(p^{k+1})$ 域上令

$$
a_j(x) \equiv w(x)b_j(x) (\bmod m(x)) \tag{2.7.3}
$$
$$
j = 1,2,\cdots,n
$$

将 $a_1(x), a_2(x), \cdots, a_n(x)$ 公开. 对于 n 位 0, 1 字符串的明文

$$
m = m_1m_2\cdots m_n
$$

加密步骤如下:

$$c(x) = m_1a_1(x) + m_2a_2(x) + m_3a_3(x) + \cdots + m_na_n(x)$$

解密办法是对于密文 $c(x)(\bmod m(x))$ 乘以 $\overline{w}(x)$，$\overline{w}(x)$ 满足

$$w(x)\overline{w}(x) \equiv 1(\bmod m(x))$$

$$\begin{aligned}\overline{w}(x)c(x) &= m_1\overline{w}(x)a_1(x) + m_2\overline{w}(x)a_2(x) \\ &\quad + \cdots + m_n\overline{w}(x)a_n(x) \\ &\equiv m_1b_1(x) + m_2b_2(x) + \cdots + m_nb_n(x) \\ &\quad (\bmod m(x))\end{aligned} \tag{2.7.4}$$

比较等式两端某 x^i 项系数便可求得 m_1，m_2，$\cdots$，m_n，这相当于解一超递增序列的背包问题.

也可以考虑分组解决. 设 $n=rs$，将 m_1，m_2，$\cdots$，m_n 分成 r 组，每组 s 个，分别由比较 r 项 x^{i_1}，x^{i_2}，$\cdots$，x^{i_r}的系数来决定. i_1，i_2，$\cdots$，i_r 可以随机选择. 这只要矩阵 B 在第 i_1，i_2，$\cdots$，i_r 列具有如下性质便可. 首先每一列有且仅有 s 个非零元素，其余的 $n-s$ 个元素均为零. 其次，若任一列中第 j 个元素非零，则其余 $r-1$ 列的第 j 个元素均为零. 即 n 个非零元素覆盖了所有 n 行，在这第 i_1，i_2，$\cdots$，i_r 列中每行有一个且仅有一个非零元素. 除这 r 列外，矩阵中其余 $k+1-r$ 列的元素实际上是多余的，可以任意选取，只要不超过规定的界即可.

如何选择素数 p 特别是不可化约的 $k+1$ 次多项式 $m(x)$，可以参考近世代数，这里不赘述. 读者可构造一个 $n=20$，$r=4$，$s=5$ 的例子. 例如取 $m(x)=x^{20}-2$，$w(x)=x^{10}+1$，$\overline{w}(x)=x^{10}-1$，并利用它进行加密和解密.

显然，这样的变形并没有改变 MH 背包公钥的实质. 但第 i_1，i_2，$\cdots$，i_r 列的任意选择，以及各列中 s 个非零超递增元素的排列的任意性，都给破译者增加了不少的麻烦. 然而付出的代价也是巨大的，即公钥的量大大地膨胀了.

§8 沙米尔(Shamir)的攻击

MH 背包公钥密码中的公钥序列 $a_1, a_2, \cdots, a_n$ 是由超递增序列 b_1，b_2，…，b_n 通过默科—赫尔曼变换得到的. 即

$$a_k = wb_k(\bmod m), \quad k = 1,2,\cdots,n$$

假定所得的 a_k，它的二进制数表示的位数是相同的，一般假定为 200 比特. 沙米尔便抓住它作突破口将它攻破了. 了解他是怎样进行分析的对我们颇有启发.

(a) 算法的非形式化叙述. 破译的问题在于寻找一对陷门 (m, u) 使得：

$$b_k \equiv ua_k(\bmod m) \quad k = 1,2,\cdots,n$$

是超递增序列. 若超递增序列的每一项大致是前面一项的两倍，最后一项比 $m/2$ 小，这样有：

$$b_n < 2^{-1}m, \ b_{n-1} < 2^{-2}m, \ \cdots, \ b_1 < 2^{-n}m$$

以 $b_1 \equiv ua_1(\bmod m)$ 为例，作图象(图 2.8.1).

$$y = a_1x(\bmod m), \quad 0 \leqslant x \leqslant m \qquad (2.8.1)$$

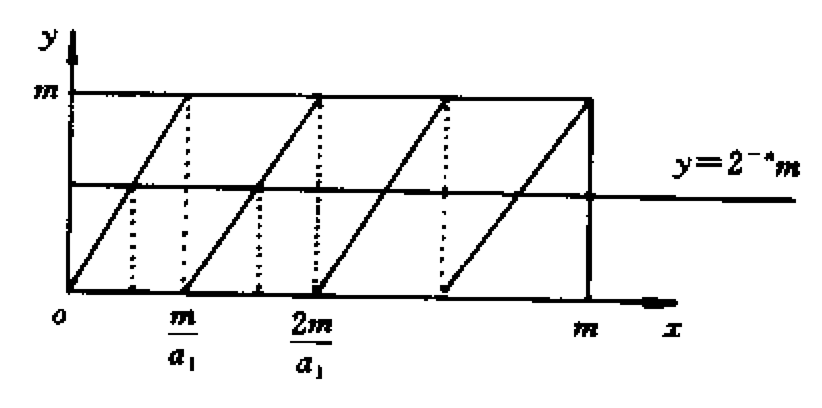


图 2.8.1

这些斜率为 a_1 的锯齿曲线的极小点分别为 $m/a_1, 2m/a_1, \cdots$，间

隔为 m/a_1. 由于

$$b_1 < 2^{-n}m \tag{2.8.2}$$

所以,u 必须在区间:

$$\left(\frac{km}{a_1},\frac{km}{a_1}+\frac{2^{-n}m}{a_1}\right) \tag{2.8.3}$$

上,$k = 0,1,2,\cdots,a_1 - 1$. 这可从图 2.8.2 看出.

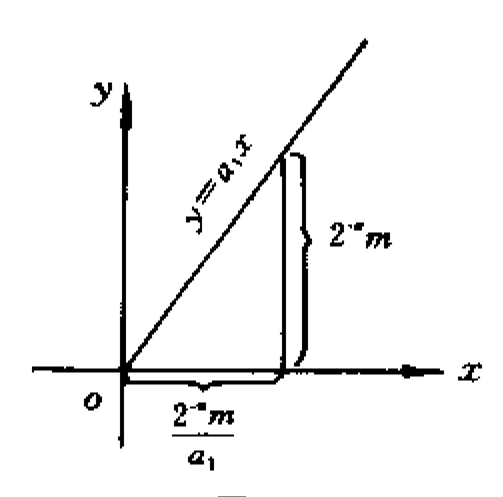


图 2.8.2

但 u 和 m 都是要求的整数. 为方便起见,将 m 作为 1 个单位,问题导至求 u,m 使比值 u/m 必须位于下列区间上:

$$\left(\frac{k}{a_1},\frac{k}{a_1}+d_1\right),\quad k = 0,1,2,\cdots,a_1 - 1 \tag{2.8.3$'$}$$

其中 $d_1 = 2^{-n}/a_1$,d_1 的几何意义见图 2.8.2.

用类似的方法讨论:

$$b_2 \equiv ua_2 \pmod m \tag{2.8.4}$$

作 $y=a_2x \pmod 1$ 的图象. 由于

$$b_2 < 2^{-n+1} \tag{2.8.5}$$

所以,$\frac{u}{m}$ 必须位于下列区间上.

$$\left(\frac{k}{a_2},\frac{k}{a_2}+d_2\right),\quad k = 0,1,2,\cdots,a_2 - 1 \tag{2.8.6}$$

其中$d_2 = 2^{-n+1}/a_2$. $\frac{u}{m}$必须同时在区间(2.8.3′)和(2.8.6)上，故必须在区间(2.8.3′)和(2.8.6)的交集上. 因此求$\frac{u}{m}$的所在范围缩小了. 而且，$\frac{u}{m}$出现在(1)和(4)的极小点十分接近的地方.

对$a_3, a_4, \cdots, a_n$用类似的算法进行讨论，可以得出所求的$\frac{u}{m}$是图象$y = a_i x \pmod 1$, $i = 1, 2, \cdots, n$极小点的聚点.

(b) 在进一步讨论沙米尔的攻击方法的其它细节之前先看一个例子有助子理解.

设已知公钥序列为{467,355,131,318,113,21,135,215}.

$a_1 = 467$，故$\frac{u}{m}$应落入下列区间上：

$$A_n = \left(\frac{n}{467}, \frac{n}{467} + \frac{1}{467 \times 256}\right), \quad n = 1, 2, \cdots, 466$$

其中$256 = 2^8$.

$a_2 = 355$，同理$\frac{u}{m}$应落入下面的区间上：

$$B_k = \left(\frac{k}{355}, \frac{k}{355} + \frac{1}{355 \times 128}\right), \quad k = 1, 2, \cdots, 354$$

因此，$\frac{u}{m}$必须在$\bigcup_{k=1}^{466} A_k$和$\bigcup_{k=1}^{354} B_k$的交集上. 通过计算得它们是下面4个区间：

$$\begin{array}{l} (0.053533190578158, 0.053541555139186) \\ (0.4732394366191971 8, 0.473241769271949) \\ (0.526766595289079, 0.526774959850107) \\ (0.580299785867238, 0.580303697183099) \end{array} \qquad (2.8.7)$$

若考虑$a_3 = 131$，$\frac{u}{m}$还必须位于以下的区间上：

$$C_j = \left(\frac{j}{131}, \frac{j}{131} + \frac{1}{131 \times 32}\right), \quad j = 1, 2, \cdots, 130$$

即$\frac{u}{m}$必须位于（2.8.7）的 4 个区间和 $\bigcup_{j=1}^{130} C_j$ 的交集上，于是再将$\frac{u}{m}$搜索区间缩至以下的 2 个：

（0.053533190578158，0.053541555139186）

（0.526766595289079，0.526774959850107）

进而考虑 $a_4 = 318$，$\frac{u}{m}$还必须落入如下的区间上：

$$D_l = \left(\frac{l}{318}, \frac{l}{318} + \frac{1}{318 \times 16}\right), \quad l = 1, 2, \cdots, 317$$

将$\frac{u}{m}$的搜索区间进一步缩小为：

（0.0535331905178158，0.053541555139186）　　（2.8.8）

如何确定整数 u 和 m 使得$\frac{u}{m}$落入（2.8.8）区间？这将在后面给出一般的方法．对于本例至少有

（i）$u=53$，$m=990$

通过变换 $b_i = 53a_i \pmod{990}$，　$i=1, 2, 3, \cdots, 8$，得超递增序列：

1，5，13，24，49，123，225，505

例如，$53 \times 467 = 24751 \equiv 1 \pmod{990}$

$53 \times 355 = 18815 \equiv 5 \pmod{990}$

等．

（ii）$u=28$，$m=523$ 得：

1，3，7，13，26，65，119，269

（c）必须取多少个关于 a_i 的锯齿曲线足以确定 u/m 的值？上例取 4 个是否有根据？现分析如下．

假定需要 l 个关于 a_i 的锯齿曲线叠加以确定比值 u/m．a_1

的锯齿曲线的极小点的横坐标为$\frac{1}{a_1}, \frac{2}{a_1}, \cdots\cdots, a_i$锯齿曲线的极小点与$k/a_1$最靠近的点必然落在区间

$$\left(\frac{k}{a_1}-\frac{1}{2a_i}, \frac{k}{a_1}+\frac{1}{2a_i}\right)$$

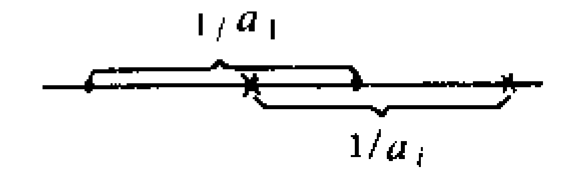


图 2.8.3

上，这一点只要观察图 2.8.3 就不难知道．假定关于a_i的锯齿曲线的极小点位置作为独立的随机变量均匀分布在上面区间上．对于其中k/a_1点是a_i锯齿曲线极小点的聚点的概率约为

$$\frac{d_i}{1/a_i} \approx \frac{2^{-n+i-1}/a_i}{1/a_i}=2^{-n+i-1}$$

设$a_2, a_3, \cdots, a_l$的锯齿曲线的极小点靠近a_1曲线的第p个极小点的概率约为

$$2^{-n+1}\cdot 2^{-n+2}\cdots\cdots 2^{-n+l-1}=2^{-ln+n+\frac{l^2}{2}}$$

a_1曲线有a_1个极小点，故聚点的期望值为

$$a_1\cdot 2^{-ln+n+l^2/2} \approx 2^{dn-ln+n+l^2/2}$$

显然至少有一个聚点，但希望计算出来的聚点越少越好，不妨假定：

$$2^{dn-ln+n+\frac{1}{2}l^2} \leqslant 1$$

$$dn-ln+n+\frac{1}{2}l^2 \leqslant 0$$

$$l \geqslant d+l+\frac{l}{2n}>d+1$$

在 MH 背包公钥系统中$d=2$，$l>3$，故$l=4$.

必须指出前面假定$a_1, a_2, \cdots, a_n$正好是超递增序列$b_1, b_2,$

…，b_n 顺序变换得来的，如果顺序更动，还应对 $C(n,4)$ 种可能逐一搜索.

(d) 综上所述可知，确定聚点导至解下列不等式问题：

$$-\varepsilon_2 \leqslant \frac{p}{a_1} - \frac{q}{a_2} \leqslant \varepsilon_2', \quad 1 \leqslant q \leqslant a_2 - 1$$

$$-\varepsilon_3 \leqslant \frac{p}{a_1} - \frac{r}{a_3} \leqslant \varepsilon_3', \quad 1 \leqslant r \leqslant a_3 - 1$$

……

其中 p，q，r，…为整数，且 $1 \leqslant p \leqslant a_1 - 1$，$\varepsilon_i$ 和 ε_i' 分别是允许对 $\frac{p}{a_1}$ 的右边和左边的偏离.

这些不等式的求解可通过§9的 L^3 算法在多项式时间内实现.

§9 L³ 算法

L³ 算法是指由 A. K. Lenstra，H. W. Lenstra 和 L. Lovasz 提出的一种基归约算法，对于背包公钥密码系统的沙米尔攻击以及后面的拉格尼阿斯—粤得尼兹科 (Lagarias—OdLyzko) 攻击，勃里克尔 (Brickell) 攻击都用到它.

(1) 设 $\boldsymbol{b}_1$，$\boldsymbol{b}_2$，…，$\boldsymbol{b}_n \in R^n$，$\boldsymbol{L}$ 为 n 维空间 $\boldsymbol{R}^n$ 的子集：

$$\boldsymbol{L} \triangleq \left\{ \sum_{i=1}^{n} z_i \boldsymbol{b}_i \,|\, z_i \in \boldsymbol{Z} \right\} = \sum_{i=1}^{n} \boldsymbol{Z}\boldsymbol{b}_i \tag{2.9.1}$$

其中 $\boldsymbol{R}$ 为实数集合，$\boldsymbol{Z}$ 为整数集合，称 $\boldsymbol{L}$ 为格，$\boldsymbol{b}_1$，$\boldsymbol{b}_2$，…，$\boldsymbol{b}_n$ 为格 $\boldsymbol{L}$ 的一组基，n 是格 $\boldsymbol{L}$ 的秩. 格 $\boldsymbol{L}$ 的行列式定义为：

$$d(\boldsymbol{L}) = |\det(\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n)| \tag{2.9.2}$$

任给 n 个线性无关的向量 $\boldsymbol{b}_1$，$\boldsymbol{b}_2$，…，$\boldsymbol{b}_n \in \boldsymbol{R}^n$，可使用下述的格

朗姆—斯密特(Gram—Schmidt)正交化过程获得一组两两正交的向量 $\boldsymbol{b}_1^*$，$\boldsymbol{b}_2^*$，…，$\boldsymbol{b}_n^*$

$$\boldsymbol{b}_1^* = \boldsymbol{b}_1 \tag{2.9.3}$$

$$\boldsymbol{b}_i^* = \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{ij} \boldsymbol{b}_j^* \qquad i > 1 \tag{2.9.4}$$

其中 $\mu_{ij} = (\boldsymbol{b}_i \cdot \boldsymbol{b}_j^*)/(\boldsymbol{b}_j^* \cdot \boldsymbol{b}_j^*) \qquad i > j$ (2.9.5)

上式中的“·”是通常定义的内积.

这里 $\boldsymbol{b}_i^*$ 是 $\boldsymbol{b}_i$ 在子空间 $\sum_{k=1}^{i-1} \boldsymbol{R}\boldsymbol{b}_k$ 的正交补空间上的投影. 显然有 $\sum_{k=1}^{i-1} \boldsymbol{R}\boldsymbol{b}_k = \sum_{k=1}^{i-1} \boldsymbol{R}\boldsymbol{b}_k^*$. 以后为了方便起见，用 $\boldsymbol{V}_i$ 表由 $\boldsymbol{b}_1$，$\boldsymbol{b}_2$，…，$\boldsymbol{b}_{i-1}$构成的子空间 $\sum_{k=1}^{i-1} \boldsymbol{R}\boldsymbol{b}_k$. $\sum_{k=1}^{i-1} \boldsymbol{R}\boldsymbol{b}_k$ 的正交补空间用 $\boldsymbol{V}_i$ 表之. 故 $\boldsymbol{b}_i^*$ 是 $\boldsymbol{b}_i$ 在 $\boldsymbol{V}_i$ 上的投影.

格 $\boldsymbol{L}$ 的一组基 $\boldsymbol{b}_1$，$\boldsymbol{b}_2$，…，$\boldsymbol{b}_n$ 如果同时满足以下两条件：

$$|\mu_{ij}| \leqslant \frac{1}{2}, \quad i > j \tag{2.9.6}$$

$$|\boldsymbol{b}_i^* + \mu_{i,i-1} \boldsymbol{b}_{i-1}^*|^2 \geqslant \frac{3}{4} |\boldsymbol{b}_{i-1}^*|^2, \quad i > 1 \tag{2.9.7}$$

则称 $\boldsymbol{b}_1$，$\boldsymbol{b}_2$，…，$\boldsymbol{b}_n$ 为格 $\boldsymbol{L}$ 的一组归约基. “| |”表示通常的欧几里得长度.

注意：这里 $\boldsymbol{b}_i^* + \mu_{i,i-1} \boldsymbol{b}_{i-1}^*$ 和 $\boldsymbol{b}_{i-1}^*$ 实际上分别是 $\boldsymbol{b}_i$ 和 $\boldsymbol{b}_{i-1}$ 在 $\boldsymbol{V}_{i-1}$ 上的投影.

例：$\boldsymbol{b}_1 = (1,1,1)$，$\boldsymbol{b}_2 = (1,1,-1)$，$\boldsymbol{b}_3 = (-1,1,1)$

令：$\boldsymbol{b}_1^* = \boldsymbol{b}_1 = (1,1,1)$

$\mu_{21} = (\boldsymbol{b}_2 \cdot \boldsymbol{b}_1^*)/(\boldsymbol{b}_1^* \cdot \boldsymbol{b}_1^*) = 1/3$

$\boldsymbol{b}_2^* = \boldsymbol{b}_2 - \mu_{21} \boldsymbol{b}_1^* = (1,1,-1) - \frac{1}{3}(1,1,1)$

$$= \frac{1}{3}(2, 2, -4)$$

$$\mu_{31} = (\boldsymbol{b}_3 \cdot \boldsymbol{b}_1^*)/(\boldsymbol{b}_1^* \cdot \boldsymbol{b}_1^*) = \frac{1}{3}$$

$$\mu_{32} = (\boldsymbol{b}_3 \cdot \boldsymbol{b}_2^*)/(\boldsymbol{b}_2^* \cdot \boldsymbol{b}_2^*) = \left(-\frac{4}{3}\right)\bigg/\left(\frac{24}{9}\right) = -\frac{1}{2}$$

$$\boldsymbol{b}_3^* = \boldsymbol{b}_3 - \mu_{31}\boldsymbol{b}_1^* - \mu_{32}\boldsymbol{b}_2^*$$

$$= (-1,1,1) - \frac{1}{3}(1,1,1) + \frac{1}{2}\left(\frac{2}{3},\frac{2}{3},\frac{4}{3}\right)$$

$$= (-1,1,0)$$

不难发现：$\boldsymbol{b}_1$，$\boldsymbol{b}_2$，$\boldsymbol{b}_3$ 还是一组归约基.

（2）下面假定 $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n$ 是格 $L \in R^n$ 的一组归约基. $\boldsymbol{b}_1^*$，$\boldsymbol{b}_2^*$，…，$\boldsymbol{b}_n^*$ 是由格朗姆—斯密特正交化过程定义的.

定理 2.9.1

(i) $|\boldsymbol{b}_j|^2 \leqslant 2^{i-1}|\boldsymbol{b}_i^*|^2$，$1 \leqslant j \leqslant i \leqslant n$ (2.9.8)

(ii) $d(L) \leqslant \prod_{i=1}^{n} |\boldsymbol{b}_i| \leqslant 2^{n(n-1)/4} d(L)$ (2.9.9)

(iii) $|\boldsymbol{b}_1| \leqslant 2^{(n-1)/4} d(L)^{1/n}$ (2.9.10)

证：(i) 由 (2.9.7) 有：

$$|\boldsymbol{b}_i^* + \mu_{i,i-1}\boldsymbol{b}_{i-1}^*|^2 = (\boldsymbol{b}_i^* + \mu_{i,i-1}\boldsymbol{b}_{i-1}^*) \cdot (\boldsymbol{b}_i^* + \mu_{i,i-1}\boldsymbol{b}_{i-1}^*)$$

$$= (\boldsymbol{b}_i^* \cdot \boldsymbol{b}_i^*) + \mu_{i,i-1}(\boldsymbol{b}_{i-1}^* \cdot \boldsymbol{b}_i^*)$$

$$+ \mu_{i,i-1}(\boldsymbol{b}_i^* \cdot \boldsymbol{b}_{i-1}^*)$$

$$+ \mu_{i,i-1}^2(\boldsymbol{b}_{i-1}^* \cdot \boldsymbol{b}_{i-1}^*)$$

$$= |\boldsymbol{b}_i^*|^2 + \mu_{i,i-1}^2|\boldsymbol{b}_{i-1}^*|^2$$

$$\geqslant \frac{3}{4}|\boldsymbol{b}_{i-1}^*|$$

$\because$ $|\mu_{i,i-1}| \leqslant \frac{1}{2}$

$\therefore$ $|\boldsymbol{b}_i^*|^2 \geqslant \left(\frac{3}{4} - \mu_{i,i-1}^2\right)|\boldsymbol{b}_{i-1}^*|^2 \geqslant \frac{1}{2}|\boldsymbol{b}_{i-1}^*|^2$

对上式进行递归可得：

$$|\boldsymbol{b}_i^*|^2 \geqslant \frac{1}{2}|\boldsymbol{b}_{i-1}^*|^2 \geqslant \frac{1}{2^2}|\boldsymbol{b}_{i-2}^*|^2 \geqslant \cdots \geqslant \frac{1}{2^l}|\boldsymbol{b}_{i-l}^*|^2$$

令　$i-l=j$，$l=i-j$.

$\therefore$　$|\boldsymbol{b}_j^*|^2 \leqslant 2^{i-j}|\boldsymbol{b}_i^*|^2$，　$1 \leqslant j \leqslant i \leqslant n$　　　　(2.9.11)

由 (2.9.4)，

$$\begin{aligned}
|\boldsymbol{b}_j|^2 &= \left|\boldsymbol{b}_j^* + \sum_{k=1}^{j-1}\mu_{jk}\boldsymbol{b}_k^*\right|^2 \\
&= |\boldsymbol{b}_j^*|^2 + \sum_{k=1}^{j-1}\mu_{jk}^2|\boldsymbol{b}_k^*|^2 \\
&\leqslant |\boldsymbol{b}_j^*|^2 + \sum_{k=1}^{j-1}\frac{1}{4}|\boldsymbol{b}_j^*|^2 \\
&\leqslant |\boldsymbol{b}_j^*|^2 + \frac{1}{4}\sum_{k=1}^{j-1}Z^{j-k}|\boldsymbol{b}_j^*|^2 \\
&= \left[1 + \frac{1}{4}(2^j - 2)\right]\cdot|\boldsymbol{b}_j^*|^2 \\
&\leqslant 2^{j-1}|\boldsymbol{b}_j^*|^2 \\
&\leqslant 2^{j-1}\cdot 2^{i-j}|\boldsymbol{b}_i^*|^2 \\
&= 2^{i-1}|\boldsymbol{b}_i^*|^2 \qquad (1\leqslant j\leqslant i\leqslant n)
\end{aligned}$$

(ii) $$\begin{aligned}
d(L) &= |\det(\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n)| \\
&= |\det(\boldsymbol{b}_1^*, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n)| \\
&= |\det(\boldsymbol{b}_1^*, \boldsymbol{b}_2^* + \mu_2\boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_n)| \\
&= |\det(\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \cdots, \boldsymbol{b}_n)| \\
&= \cdots \\
&= |\det(\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \cdots, \boldsymbol{b}_n^*)|
\end{aligned}$$

由于 $\boldsymbol{b}_1^*$，$\boldsymbol{b}_2^*$，…，$\boldsymbol{b}_n^*$ 是两两正交，故

$$d(L) = \prod_{i=1}^{n}|\boldsymbol{b}_i^*|$$

由于 $\boldsymbol{b}_i^*$ 是在 $\boldsymbol{V}_i$ 上的投影，所以 $|\boldsymbol{b}_i^*| \leqslant |\boldsymbol{b}_i|$.

$$\therefore \quad d(L) = \prod_{j=1}^{n} |\boldsymbol{b}_j^*| \leqslant \prod_{j=1}^{n} |\boldsymbol{b}_j|$$

由（2.9.8）得：

$$|\boldsymbol{b}_j|^2 \leqslant 2^{j-1} |\boldsymbol{b}_j^*|^2$$

$$\therefore \quad \prod_{j=1}^{n} |\boldsymbol{b}_j|^2 \leqslant \prod_{j=1}^{n} 2^{j-1} |\boldsymbol{b}_j^*|^2$$

$$= 2^{\sum_{i=1}^{n}(i-1)} \prod_{j=1}^{n} |\boldsymbol{b}_j^*|^2$$

$$= 2^{\frac{n(n-1)}{2}} d(L)^2$$

$$\therefore \quad \prod_{j=1}^{n} |\boldsymbol{b}_j| \leqslant 2^{n(n-1)/4} d(L)$$

（iii）在（2.9.8）中令 $j=1$ 得：

$$|\boldsymbol{b}_1|^2 \leqslant 2^{i-1} |\boldsymbol{b}_i^*|^2$$

$$\prod_{i=1}^{n} |\boldsymbol{b}_1|^2 = |\boldsymbol{b}_1|^{2n} \leqslant \prod_{i=1}^{n} 2^{i-1} |\boldsymbol{b}_i^*|^2 = 2^{n(n-1)/2} d(L)^2$$

$$\therefore \quad |\boldsymbol{b}_1| \leqslant 2^{(n-1)/4} d(L)^{1/n}$$

定理 2.9.2 $\forall\, x \in L$，$x \neq 0$，则

$$|\boldsymbol{b}_1|^2 \leqslant 2^{n-1} |x|^2 \tag{2.9.12}$$

证：$\because \quad x \in L$

$$\therefore \quad x = \sum_{i=1}^{n} z_i \boldsymbol{b}_i = \sum_{i=1}^{n} z_i^* \boldsymbol{b}_i^*, z_i \in \boldsymbol{Z}, z_i^* \in \boldsymbol{R}$$

令 i 是使 $z_i \neq 0$ 的最大下标．可证 $z_i = z_i^*$．因为

$$x = \sum_{j=1}^{i} z_j \boldsymbol{b}_j$$

$$= z_i \boldsymbol{b}_i + \sum_{j=1}^{i-1} z_j \boldsymbol{b}_j$$

$$= z_i \left(\boldsymbol{b}_i^* + \sum_{j=1}^{i-1} \mu_{ij} \boldsymbol{b}_j^* \right) + \sum_{j=1}^{i-1} z_i^* \boldsymbol{b}_i^*$$

$$= z_i \boldsymbol{b}_i^* + \sum_{j=1}^{i-1} (\mu_{ij} z_i + z_j^*) \boldsymbol{b}_j^*$$

$\therefore \quad z_i = z_i^*$

$$\therefore \quad |x|^2 \geqslant z_i^2 |\boldsymbol{b}_i^*|^2 \geqslant |\boldsymbol{b}_i^*|^2 \qquad (2.9.13)$$

由（2.9.8）得：

$$|\boldsymbol{b}_1|^2 \leqslant 2^{i-1} |\boldsymbol{b}_i^*|^2 \leqslant 2^{n-1} |\boldsymbol{b}_i^*|^2 \leqslant 2^{n-1} |x|^2$$

定理 2.9.3 $x_1, x_2, \cdots, x_t \in \boldsymbol{L}$ 是 t 个线性无关的向量，$1 \leqslant t \leqslant n$，则：

$$|\boldsymbol{b}_j|^2 \leqslant 2^{n-1} \max\{|x_1|^2, |x_2|^2, \cdots, |x_t|^2\} \quad 1 \leqslant j \leqslant t$$

证：$\because \quad x_j \in \boldsymbol{L}$

$$\therefore \quad x_j = \sum_{i}^{n} z_{ij} \boldsymbol{b}_i \quad j = 1, 2, \cdots, t$$

对确定的 j，令 $i(j)$ 表示使 $z_{ij} \neq 0$ 的最大下标. 由（2.9.13）得：

$$|x_j|^2 \geqslant |\boldsymbol{b}_{i(j)}^*|^2$$

将 x_j 重新排列使得：

$$i(1) \leqslant i(2) \leqslant \cdots \leqslant i(t)$$

则有 $j \leqslant i(j), j = 1, 2, \cdots, t$

否则若存在 k，$k > i(k)$，则 $x_1, x_2, \cdots, x_k$ 都在 $\sum_{k=1}^{i(k)} \boldsymbol{R}\boldsymbol{b}_k$ 中，这和假定 $x_1, x_2, \cdots, x_k$ 线性无关相矛盾. 由(2.9.8)和(2.9.13)得：

$$|\boldsymbol{b}_j|^2 \leqslant 2^{n-1} |\boldsymbol{b}_{i(j)}|^2 \leqslant 2^{n-1} |\boldsymbol{b}_{i(j)}^*|^2 \leqslant 2^{n-1} |x_j|^2, \quad 1 \leqslant j \leqslant t$$

(3) 下面介绍一种从格 $\boldsymbol{L}$ 的一组基 $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n$ 开始，构造一组归约基的算法，也就是所谓的 $\boldsymbol{L}^3$ 算法.

首先，利用格朗姆—斯密特(Gram-Schmidt)正交化方法计算得 $\boldsymbol{b}_j^*$ 和 μ_{ij}，$1 \leqslant i \leqslant n, 1 \leqslant j < i$. 然后开始一个归约过程.

在这个归约过程中 $b_1,b_2,\cdots,b_n$ 可能改变，然而始终保持是 L 的一组基. 如果某个 b_i 发生变化，则相应地 b_i^* 和 μ_{ij} 也改变，但(2.9.4)和(2.9.5)依然成立，每次归约的结果总要使得一部分基满足下面两个条件：

$$|\mu_{ij}|\leqslant\frac{1}{2},\ 1\leqslant j<i\leqslant k \tag{2.9.14}$$

$$|b_i^*+\mu_{i,i-1}b_{i-1}^*|^2\geqslant\frac{3}{4}|b_{i-1}^*|^2\quad 1<i<k \tag{2.9.15}$$

从 $k=2$ 开始，上述条件自然成立. 若 $k=n+1$，则归约过程终止，便获得一组归约的基 $b_1,b_2,\cdots,b_n$.

$k>1$ 时，若 $|\mu_{k,k-l}|\leqslant\frac{1}{2}$ 不成立，令 r 是与 $\mu_{k,k-1}$ 最接近的整数，用 $b_{k-r}b_{k-l}$ 来取代 b_k，对 $j<k-l$ 用 $\mu_{kj}-r\mu_{k-l,j}$ 代替 μ_{kj}，用 $\mu_{k\,k-r}-r$ 代替 $\mu_{k,k-1}$，其它的 μ_{ij} 和 b_i^* 不变，使 $|\mu_{k\,k-l}|\leqslant\frac{1}{2}$. 在归约的过程中，每一步都可能遇上两种情况：

(I) $k\geqslant2$ 且 $|b_k^*+\mu_{k\,k-1}b_{k-1}^*|^2<\frac{3}{4}|b_{k-1}^*|^2$

这时令 b_k 和 b_{k-1} 互换，其它的 b_i 保持不变. 重新计算 b_{k-1}^*，b_k^*，$\mu_{k\,k-1}$，$\mu_{k-1\,j}$，μ_{kj}，$\mu_{i\,k-1}$，μ_{ik}. 其中 $j>k-1$，$i>k$. 注意：b_{k-1}^* 和 $b_k^*+\mu_{k\,k-1}b_{k-1}^*$ 也互换，则有 b_{k-1}^* 小于原来的 b_{k-1}^* 的 $\frac{3}{4}$，然后将 k 减 1，继续进行迭代.

(II) $k=1$ 或 $|b_k^*+\mu_{k\,k-1}b_{k-1}^*|^2\geqslant\frac{3}{4}|b_{k-1}^*|^2$.

这时先查出有 $\mu_{kl}>\frac{1}{2}$，令 b_k 用 $b_{k-r}b_l$ 取代之，r 是最接近 μ_{kl} 的整数. 反复进行，直到所有的 μ_{kl} 都满足条件为止，然后 k 加 1.

下面给出全部算法：其中 $B_i=(b_i^*\cdot b_i^*)$

$$
\left.\begin{array}{l}
\boldsymbol{b}_i^* := \boldsymbol{b}_i; \\
\left.\begin{array}{l}
\mu_{ij} := \boldsymbol{b}_i \cdot \boldsymbol{b}_j^* / B_j \\
\boldsymbol{b}_i^* := \boldsymbol{b}_i - \mu_{ij}\boldsymbol{b}_j^*
\end{array}\right\} \text{for } j := 1 \text{ to } i-1 \\
B_i := \boldsymbol{b}_i^* \cdot \boldsymbol{b}_i^*
\end{array}\right\} \text{for } i := 1 \text{ to } n
$$

$k := 2$

(I) for $l := k-1$　执行（*）

　if $B_k < \left(\frac{3}{4} - \mu_{k\,k-1}^2\right) B_{k-1}$, go to II;

　for $l := k-2$　downto 1　执行（*）;

　if $k = n$ 终止.

　$k := k+1$; goto (I)

(II) $\mu := \mu_{k\,k-1}$;　$B := B_k + \mu^2 B_{k-1}$;　$\mu_{k\,k-1} := \mu B_{k-1}/B_j$

　$B_k := B_{k-1}B_k/B$;　$B_{k-1} := B$;

$$
\begin{pmatrix} \boldsymbol{b}_{k-1} \\ \boldsymbol{b}_k \end{pmatrix} := \begin{pmatrix} \boldsymbol{b}_k \\ \boldsymbol{b}_{k-1} \end{pmatrix};
$$

$$
\begin{pmatrix} \mu_{k-1\,j} \\ \mu_{kj} \end{pmatrix} := \begin{pmatrix} \mu_{kj} \\ \mu_{k-1\,j} \end{pmatrix} \quad \text{for } j := 1 \text{ to } k-2
$$

$$
\begin{pmatrix} \mu_{i\,k-1} \\ \mu_{ik} \end{pmatrix} := \begin{pmatrix} 1 & \mu_{k\,k-1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -\mu \end{pmatrix} \begin{pmatrix} \mu_{i\,k-1} \\ \mu_{ik} \end{pmatrix}
$$

　if $k > 2$ then $k := k-1$ goto (I)

　（*）if $|\mu_{kl}| > \frac{1}{2}$　then

$$
\begin{cases}
r := \text{离 } \mu_{kl} \text{ 最近的整数}, \\
\boldsymbol{b}_k := \boldsymbol{b}_k - r\boldsymbol{b}_{k-1}; \\
\mu_{kj} := \mu_{kj} - r\mu_{lj} \quad \text{for } j := 1 \text{ to } l-1 \\
\mu_{kl} := \mu_{kl} - r
\end{cases}
$$

（4）在 §8 中已将对 MH 背包公钥系统的攻击归结为解一组线性不等式：

$$-\varepsilon_2 \leqslant \frac{p}{a_1} - \frac{q}{a_2} \leqslant \varepsilon_2{}', \quad \begin{array}{l} 1 \leqslant p \leqslant a_1 - 1 \\ 1 \leqslant q \leqslant a_2 - 1 \end{array}$$

$$-\varepsilon_2 \leqslant \frac{p}{a_1} - \frac{r}{a_3} \leqslant \varepsilon_3{}', \quad 1 \leqslant r \leqslant a_3 - 1$$

$$\cdots\cdots \qquad \cdots\cdots$$

可将上述不等式转化为

$$|p_i - q\omega_i| \leqslant \varepsilon, \qquad 1 \leqslant i \leqslant n$$

$$1 \leqslant q \leqslant 2^{n(n+1)/4} \varepsilon^{-n}$$

其中 ω_i 是有理数，$p_1, p_2, \cdots, p_n, q$ 是整数，$0 < \varepsilon < 1$，可令：

$$b_1 = (1, 0, \cdots, 0, -\omega_1),$$

$$b_2 = (0, 1, \cdots, 0, -\omega_2),$$

$$\cdots\cdots$$

$$b_n = (0, 0, \cdots, 1, -\omega_n),$$

$$b_{n+1} = (0, 0, \cdots, 0, 2^{-n(n+1)/4} \varepsilon^{n+1})$$

通过 L^3 算法可得一归约基 $b_1^*, b_2^*, \cdots, b_{n+1}^*$. 由定理 2.9.1 可知：

$$|b_1^*|^2 \leqslant 2^{n/4} d(L)^{1/(n+1)}$$

$$d(L) = \begin{vmatrix} 1 & 0 & & & 0 & -\omega_1 \\ 0 & 1 & \ddots & & 0 & -\omega_2 \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & & \ddots & 1 & -\omega_n \\ 0 & 0 & & & 0 & 2^{-(n+1)n/4} \varepsilon^{n+1} \end{vmatrix}$$

$$= 2^{-n(n+1)/4} \varepsilon^{n+1}$$

$$\therefore \quad 2^{n/4} d(L)^{1/(n+1)} = 2^{n/4} (2^{-n(n+1)/4} \varepsilon^{n+1})^{1/(n+1)}$$

$$= 2^{n/4} 2^{-n/4} \cdot \varepsilon = \varepsilon$$

由于 $b_1^* \in L$，故

$$b_1^* = \sum_{i=1}^{n} p_i b_i + q b_{n+1}, \qquad p_i, q \in Z$$

$\therefore \quad b_1^* = (p_1 - q\omega_1, p_2 - q\omega_2, \cdots, p_n - q\omega_n, q2^{-n(n+1)/4}\varepsilon^{n+1})$

$\therefore \quad |p_i - q\omega_i| \leqslant \varepsilon, \quad i = 1, 2, \cdots, n$

$|q2^{-n(n+1)/4}\varepsilon^{n+1}| < \varepsilon$

$|q| \leqslant 2^{n(n+1)/4}\varepsilon^{-n}$

最后有两种可能：q 或是正整数，或是负的．若是后者，b_1^* 用 $-b_1^*$ 代之．

§10　拉格尼阿斯—粤得尼兹科(Lagarias-Odlyzko)和勃里克尔(Brickell)的攻击

沙米尔攻击方法发表之后，默科和赫尔曼试图重复利用他们构造公开的背包序列的 MH 算法，企图使超递增序列的"原形"隐蔽得更深一些，使沙米尔的攻击失败．但过了两年之后，拉格尼阿斯—粤得尼兹科和勃里克尔几乎同时提出攻击方法，虽然出发点各异，但结论非常接近，可谓殊途同归．

首先定义背包序列 $\{a_1, a_2, \cdots, a_n\}$ 的背包密度：

$$D = n/\log_2(\max_i a_i)$$

这里 n 实际是明文 $m = m_1m_2\cdots m_n$ 的比特数．

拉格尼阿斯—粤得尼兹科和勃里克尔都证明了对 $D < 0.645$ 的背包公钥码问题可以破译．

背包密度可以近似地看作是

$$D \approx \frac{\text{明文的比特数}}{\text{密文的平均比特数}}$$

例如，$n = 100$, $\log_2(\max_i a_i) = 200$，则 $D = 0.5$

一般说来信息率越高，陷井信息越难隐蔽．所谓信息率就是明文长度和码文长度之比，所以攻破 $D < 0.645$ 的背包公钥系统的方法较之沙米尔方法更具一般性．

下面仅叙述拉格尼阿斯—粤得尼兹科方法．假设已知公开的背包序列 $a_1, a_2, \cdots, a_n$，及 b，$a_i \in \mathbf{Z}$，$i=1, 2, \cdots, n$，求解：

$$\sum_{k=1}^{n} a_k x_k = b, \quad x_i = 0.1, \quad i = 1,2,\cdots,n \qquad (2.10.1)$$

S_1. 取格 L 的一组基：

$\boldsymbol{b}_1 = (1,0,0,\cdots,0,-a_1)$

$\boldsymbol{b}_2 = (0,1,0,\cdots,0,-a_2)$

……

$\boldsymbol{b}_n = (0,0,0,\cdots,0,-a_n)$

$\boldsymbol{b}_{n+1} = (0,0,0,\cdots,0,b)$

S_2. 利用 L^3 算法求 L 的归约基 $\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \cdots, \boldsymbol{b}_n^*, \boldsymbol{b}_{n+1}^*$.

S_3. 若任意 $\boldsymbol{b}_i^* = (b_{i,1}^*, b_{i,2}^*, \cdots, b_{i,n+1}^*)$ 有所有 $b_{ij}^* = 0$ 或常数 $\lambda, 1 \leqslant j \leqslant n$,

$$x_j = \frac{1}{\lambda} b_{i,j}^*, \qquad i \leqslant j \leqslant \lambda$$

给出(2.10.1)的解，则停止．否则转 S_4.

S_4. $b = \sum_{i=1}^{n} a_i - b$，重复 S_1 到 S_3，得解$(x_1 x_2 \cdots, x_n)$，原来背包问题的解为$(1 - x_1, 1 - x_2, \cdots, 1 - x_n)$.

例 已知超递增序列

$b_1=128$，$b_2=143$，$b_3=275$，$b_4=562$，$b_5=1156$，

$b_6=2531$，$b_7=6305$，$b_8=11590$，$b_9=23145$，

$b_{10}=50308$，$b_{11}=100697$，$b_{12}=239213$

取 $m=978426$，$w=4865$，将 $\{b_i\}$ 转换为

$a_1=00622720$，$a_2=00695695$，$a_3=00359449$

$a_4=00777278$，$a_5=00731810$，$a_6=00572203$

$a_7=00342619$，$a_8=00517768$，$a_9=00081435$

$a_{10}=00141920$，$a_{11}=00677905$，$a_{12}=00422731$

取格 L 的一组基如下：

$\boldsymbol{b}_1=(1,0,0,0,0,0,0,0,0,0,0,0-622720)$

$\boldsymbol{b}_2=(0,1,0,0,0,0,0,0,0,0,0,0-695695)$

$\boldsymbol{b}_3=(0,0,1,0,0,0,0,0,0,0,0,0-359449)$

$\boldsymbol{b}_4=(0,0,0,1,0,0,0,0,0,0,0,0-777278)$

$\boldsymbol{b}_5=(0,0,0,0,1,0,0,0,0,0,0,0,0-731810)$

$\boldsymbol{b}_6=(0,0,0,0,0,1,0,0,0,0,0,0,0-572203)$

$\boldsymbol{b}_7=(0,0,0,0,0,0,1,0,0,0,0,0-342619)$

$\boldsymbol{b}_8=(0,0,0,0,0,0,0,1,0,0,0,0-517768)$

$\boldsymbol{b}_9=(0,0,0,0,0,0,0,0,1,0,0,0-81435)$

$\boldsymbol{b}_{10}=(0,0,0,0,0,0,0,0,0,1,0,0-141920)$

$\boldsymbol{b}_{11}=(0,0,0,0,0,0,0,0,0,0,1,0-677905)$

$\boldsymbol{b}_{12}=(0,0,0,0,0,0,0,0,0,0,0,1-422731)$

$\boldsymbol{b}_{13}=(0,0,0,0,0,0,0,0,0,0,0,0\quad 2764551)$

其中 $b=2764551$

用 L^3 算法得归约基：

$\boldsymbol{b}_1^*=(-1,\ 0,-2,\ 1,\ 1,\ 0,-2,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0)$

$\boldsymbol{b}_2^*=(\ 1,\ 0,\ 3,\ 0,-1,-2,-1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0)$

$\boldsymbol{b}_3^*=(-1,\ 1,-2,\ 3,-1,\ 0,\ 0,-2,\ 1,\ 0,\ 0,\ 0,\ 0)$

$\boldsymbol{b}_4^*=(-1,-2,\ 1,\ 0,\ 0,\ 0,-1,\ 2,\ 0,\ 2,\ 1,\ 0,-1)$

$\boldsymbol{b}_5^*=(-1,-1,\ 1,\ 1,-1,\ 1,\ 0,\ 0,-1,\ 0,\ 0,\ 1,-1)$

$\boldsymbol{b}_6^*=(\ 1,-1,\ 0,-1,-1,\ 2,\ 1,\ 1,\ 0,\ 0,\ 0,-1,\ 1)$

$\boldsymbol{b}_7^*=(\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 1,\ 0)$

$\boldsymbol{b}_8^*=(\ 0,\ 3,-2,\ 1,\ 0,\ 1,\ 0,-1,\ 0,\ 1,\ 0,\ 1,\ 0)$

$\boldsymbol{b}_9^*=(\ 2,\ 0,\ 2,-1,\ 1,\ 1,-2,\ 0,\ 0,-1,\ 1,\ 1,\ 0)$

$b_{10}^{*}=(\ \ 0,-2,-2,\ \ 1,\ \ 1,-1,\ \ 1,-2,-1,\ \ 2,-1,-1,\ \ 0)$

$b_{11}^{*}=(\ \ 1,-2,-1,-1,\ \ 0,\ \ 0,\ \ 0,\ \ 1,-1,-1,\ \ 3,-1,\ \ 0)$

$b_{12}^{*}=(\ \ 1,\ \ 0,-1,-1,-2,\ \ 1,\ \ 0,\ \ 0,\ \ 2,\ \ 1,\ \ 1,\ \ 1,-2)$

$b_{13}^{*}=(\ \ 0,-1,-1,\ \ 1,\ \ 3,\ \ 1,\ \ 1,\ \ 1,-2,\ \ 0,\ \ 0,-1,-2)$

从 b_7^* 可知其分量或等于 0 或等于常数 1.

令 $x_1=x_4=x_7=x_8=x_9=x_{12}=1$

$x_2=x_3=x_5=x_6=x_{10}=x_{11}=x_{12}=0$

代入验证可得：

$$a_1+a_4+a_7+a_8+a_9+a_{12}=2764551$$

满足要求. 所以 100100111001 是对应于密文 2764551 的明文.

§ 11　椭圆曲线公钥密码

(1) 椭圆曲线的研究来源于椭圆积分：

$$\int \frac{dx}{\sqrt{E(x)}}$$

这里 $E(x)$ 是 x 的三次多项式或四次多项式. 这样的积分不能用初等函数来表达，为此引进了所谓椭圆函数. 怎样从椭圆积分引出椭圆曲线在此不多介绍. 所谓椭圆曲线指的是由韦尔斯特拉斯（Weierstrass）方程：

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6 \quad (2.11.1)$$

所确定的平面曲线. 若 F 是一个域，$a_i\in F$，$i=1, 2, 3, \cdots, 6$. 满足 (2.11.1) 的数偶 (x, y) 称为 F 域上的椭圆曲线 E 的点. F 域可以是有理数域，也可以是复数域，还可以是有限域 $GF(p^r)$. 椭圆曲线通常用 E 表示. 除了曲线 E 的所有的点外，尚需加上一个叫做无穷远点的特殊点 O. 若令

$$x=\frac{X}{Z},\qquad y=\frac{Y}{Z}$$

以之代入（2.11.1）得：

$$\left(\frac{Y}{Z}\right)^2+a_1\left(\frac{XY}{Z^2}\right)+a_3\frac{Y}{Z}=\left(\frac{X}{Z}\right)^3+a_2\left(\frac{X}{Z}\right)^2+a_4\left(\frac{X}{Z}\right)+a_6$$

$Z\neq 0$ 时，整理得：

$$Y^2Z+a_1XYZ+a_3YZ^2=X^3+a_2X^2Z+a_4XZ^2+a_6Z^3 \tag{2.11.2}$$

显然 (x,y) 和 (X,Y,Z) 相对应．而且任取一非零常数 λ，使得 (X,Y,Z) 和 $(\lambda X,\lambda Y,\lambda Z)$ 表达同一个点．但若 $Z=0$ 时，以 $(0,1,0)$ 为例，它可以看作是 $(0,1,\varepsilon)$ 当 $\varepsilon\to 0$ 而得到的．可以看作是沿 y 轴趋向无穷远．

（2）对于三次方程 $x^3+c_1x^2+c_2x+c_3=0$，令 $x=y-\frac{1}{3}c_1$ 代入得 $y^3+py+q=0$，其中 $p=c_2-\frac{1}{3}c_1^2$，$q=c_3-\frac{1}{3}c_1c_2+\frac{2}{27}c_1^3$．则三个根分别是：

$$y_1=\sqrt[3]{-\frac{1}{2}q+\sqrt{R}}+\sqrt[3]{-\frac{1}{2}q-\sqrt{R}}$$

$$y_2=\omega\sqrt[3]{-\frac{1}{2}q+\sqrt{R}}+\omega^2\sqrt[3]{-\frac{1}{2}q-\sqrt{R}}$$

$$y_3=\omega^2\sqrt[3]{-\frac{1}{2}q+\sqrt{R}}+\omega\sqrt[3]{-\frac{1}{2}q-\sqrt{R}}$$

其中 $R=\frac{1}{4}q^2+\frac{1}{27}p^3$，

$$\omega=-\frac{1}{2}+\frac{1}{2}\sqrt{-3},\qquad \omega^2=-\frac{1}{2}-\frac{1}{2}\sqrt{-3}$$

以后称 $27q^2+4p^3$ 为判别式，记为 Δ．显然

(a) $\Delta>0$ 时有一个实根和一对复根；

(b) $\Delta=0$ 时有三个实根；特别当 $\left(\frac{1}{2}q\right)^2=-\left(\frac{1}{3}p\right)^3\neq 0$ 时，三个实根中有两个相等；$p=q=0$ 时有三重零根.

(c) $\Delta<0$ 时，有三个不等的实根.

若在 (2.11.1) 中令 y 代以 $(y-a_1x-a_3)/2$，便得

$$y^2=4x^3+b_2x^2+2b_4x+b_6 \tag{2.11.3}$$

其中 $b_2=a_1^2+4a_2$， $b_4=2a_4+a_1a_3$， $b_6=a_3^2+4a_6$

(2.11.3) 的曲线关于 x 轴对称.

下面举例说明它.

例 1 $y^2=x^3-3x+3$

$y=0$ 时， $x^3-3x+3=0$， $p=-3$， $q=3$

$\Delta=27q^2+4p^3=27\times 9-4\times 27=5\times 27>0$

故有一实根，一对复根. 图象如图 2.11.1 所示.

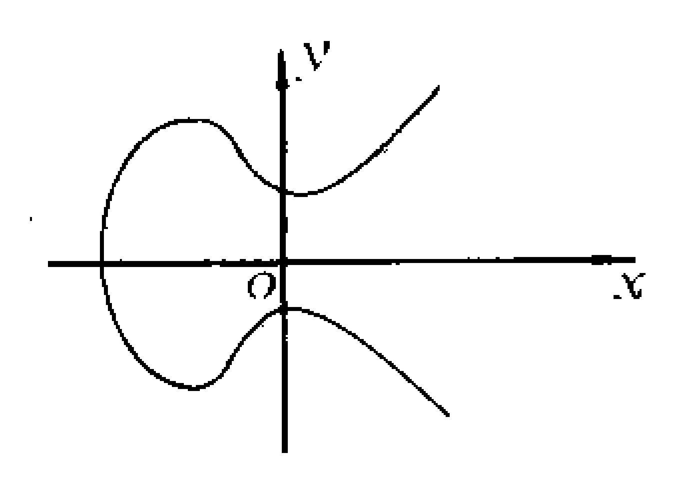


图 2.11.1

例 2 $y^2=x^3+x$.

$p=1$， $q=0$， $\Delta=4p^2=4>0$

图象如图 2.11.2 所示.

例 3 $y^2=x^3-x$， $p=-1$， $q=0$， $\Delta=-4<0$，$x^3-x=0$ 有 3 个实根，图象如图 2.11.3 所示.

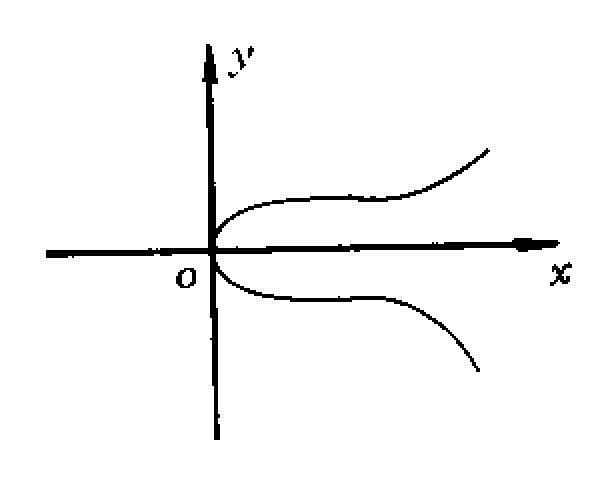


图 2.11.2

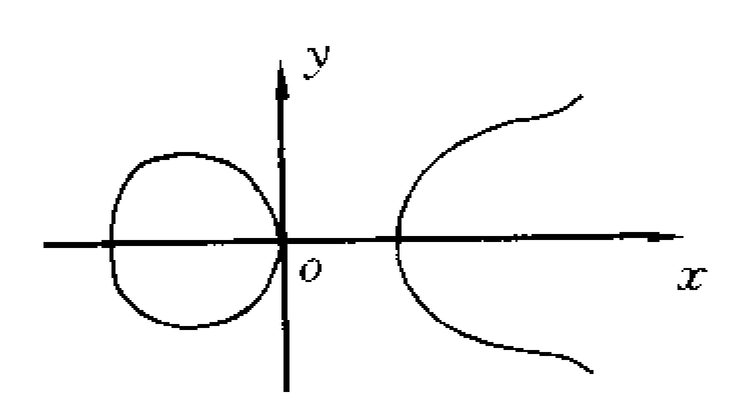


图 2.11.3

例 4　　$y^2=x^3+x^2$

$c_1=1$，　$c_2=c_3=0$，　$p=-\frac{1}{3}$，　$q=\frac{2}{27}$

$\Delta=27q^2+4p^3=4/27-4/27=0$，即 $x^3+x^2=0$ 有三个实根，两个根相等，图象见图 2.11.4.

(3) 设 $P=(x_1, y_1)$，$Q=(x_2, y_2)$ 是 E 上任意两点，L 是 PQ 连线. 若 P 和 Q 重合于一点，即 $P=Q$，则 L 便退化为 P 点的切线. 设 L 和曲线相交于另一点 R，L' 是 R 点和无穷远点 O 的连线，也就是说 L' 是过 R 点引 y 轴平行线，L' 和曲线交于一点，用 $P\oplus Q$ 表之. 实际上点 $P\oplus Q$ 和点 R 关于 x 轴对称. 曲

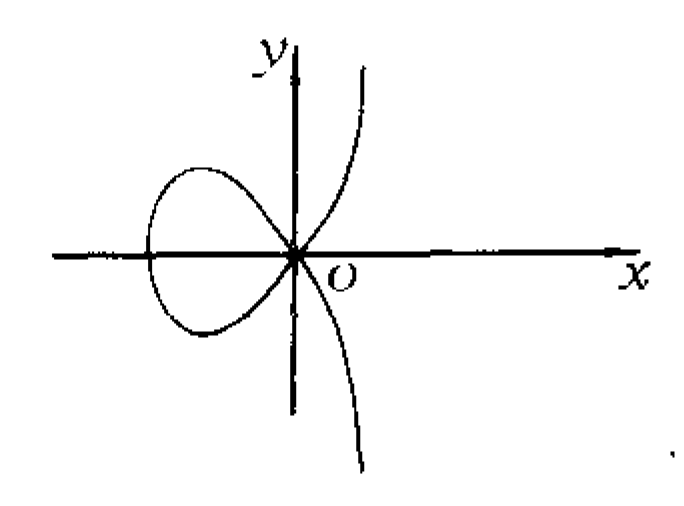


图 2.11.4

线 E 关于 x 对称，这从（2.11.3）式只含 y^2 项可知.

如若 P 和 Q 关于 x 轴对称或重合于 x 轴，则 PQ 垂直于 x 轴，这时 L 和椭圆曲线交于无穷远点 O.

例 5 椭圆曲线 E 为：

$$y^2=x^3+a_4x+a_6 \tag{2.11.4}$$

$$P=(x_1,y_1),\quad Q=(x_2,y_2),$$

过 P 和 Q 点的直线 L 设为：

$$y=mx+b$$

其中：

$$m=\frac{y_2-y_1}{x_2-x_1},\qquad b=y_1-mx_1$$

以此代入曲线 E 的方程得：

$$(mx+b)^2=x^3+a_4x+a_6$$

即 $$x^3-(mx+b)^2+a_4x+a_6=0$$

由于 x_1、x_2 是它的两个根，若 PQ 直线交曲线 E 于点 $R=(x_3,\ y_3)$，根据根与系数关系，

$$x_1+x_2+x_3=m^2$$

$\therefore \quad x_3=m^2-x_1-x_2$

$\quad\quad y_3=mx_3+b$

若 $\quad P\oplus Q=(x^*,y^*)=(x_3,-y_3)=(x_3,-(mx_3+b))$

即

$$\begin{cases} x^*=x_3=\left(\dfrac{y_2-y_1}{x_2-x_1}\right)^2-x_1-x_2 \\ y^*=-y_1+\left(\dfrac{y_2-y_1}{x_2-x_1}\right)(x_1-x_3) \end{cases} \quad (2.11.5)$$

(2.11.5)给出已知曲线(2.11.4)上两点 $P=(x_1,y_1)$，$Q=(x_2,y_2)$，求 $P\oplus Q=(x^*,y^*)$的坐标.

若 $P=Q$，对 $y^2=x^3+a_4x+a_6$，求 x 的导数得：

$$2y\frac{dy}{dx}=3x^2+a_4$$

$$\frac{dy}{dx}=\frac{3x^2+a_4}{2y}$$

$$y'|_{x=x_1}=\frac{3x_1^2+a_4}{2y_1}$$

其中 $\quad b=y_1-mx_1=y_1-\left(\dfrac{3x_1^2+a_4}{2y_1}\right)x_1$

代入曲线 E 的方程得：

$$\left(\frac{3x_1^2+a_4}{2y_1}x+b\right)^2=x^3+a_4x+b$$

根据根与系数的关系有：

$$x_1+x_2+x_3=2x_1+x_3=\left(\frac{3x_1^2+a_4}{2y_1}\right)^2$$

$$\therefore \quad x_3=\left(\frac{3x_1^2+a_4}{2y_1}\right)^2-2x_1$$

故得 $P=Q=(x_1,\ y_1)$ 时，$P\oplus Q=\ (x^*,\ y^*)$.

$$\begin{cases} x^* = \left(\dfrac{3x_1^2+a_4}{2y_1}\right)^2 - 2x_1 \\ y^* = -y_1 - \left(\dfrac{3x_1^2+a_4}{2y_1}\right)(x_3 - x_1) \end{cases} \tag{2.11.6}$$

(2.11.6) 给出当 $P=Q$ 时，求 $P\oplus Q=(x^*, y^*)$ 的公式.

(3) 对于一般的韦尔斯特拉斯方程：

$$F(x,y) = y^2 + a_1yx + a_3y - (x^3 + a_2x^2 + a_4x + a_6) = 0 \tag{2.11.7}$$

也可以得出类似的结果. 设 $P=(x_1,y_1)$, $Q=(x_2,y_2)$，过点 P 和 Q 的直线 L 为：

$$y=mx+b.$$

当 $x_1\neq x_2$ 时，$m=\dfrac{y_2-y_1}{x_2-x_1}$，$b=y_1-mx_1=y_1-\dfrac{y_2-y_1}{x_2-x_1}x_1$ 以之代入韦尔斯特拉斯方程得：$F(x,mx+b)=0$.

若直线 L 交曲线 E 于第三点 $R=(x^*,y^*)$，则 $F(x^*,y^*)=0$，令：

$$F(x,mx+b)=-(x_1-x_2)(x-x_2)(x-x^*)$$

比较系数得：

$$x_1+x_2+x^*=m^2+a_1m-a_2$$

故：

$$\begin{cases} x^*=m^2+a_1m-a_2-x_1-x_2 \\ y^*=-(m+a)x^*-m-a_3 \end{cases} \tag{2.11.8}$$

(2.11.8)给出已知曲线(2.11.7)上两点 $P(x_1,y_1)$, $Q(x_2,y_2)$，求 $P\oplus Q=(x^*, y^*)$ 的公式.

又当 $P=Q=(x_1, y_1)$ 时，有

$$2y\frac{dy}{dx}+a_1y+a_1x\frac{dy}{dx}+a_3\frac{dy}{dx}=3x^2+2a_2x+a_4$$

$$\therefore\quad \frac{dy}{dx}=\frac{3x^2+2a_2x+a_4-a_1y}{2y+a_1x+a_3} \tag{2.11.9}$$

(2.11.9) 给出曲线(2.11.7) 上点 $P(x_1, y_1)$ 的切线斜率. 设切于点 P 的切线方程为:

$$y = mx + b \tag{2.11.10}$$

则:

$$m = \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} \tag{2.11.11}$$

$$b = y_1 - mx_1 = y_1 - \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}x_1$$

$$= \frac{2y_1^2 + 2a_1x_1y_1 + a_3y_1 - 3x_1^2 - 2a_2x_1^2 - a_4x_1}{2y_1 + a_1x_1 + a_3}$$

因 $y_1^2 + a_1x_1y_1 + a_3y_1 = x_1^3 + a_2x_1^2 + a_4x_1 + a_6$

$$\therefore\quad b = \frac{2(x_1^3 + a_2x_1^2 + a_4x_1 + a_6) - 3x_1^2 - 2a_2x_1^2 - a_4x_1 - a_3y_1}{2y_1 + a_1x_1 + a_3}$$

$$= \frac{-x_1^3 + a_4x_1 + 2a_6 - a_3y_1}{2y_1 + a_1x_1 + a_3} \tag{2.11.12}$$

和前面类似的方法可求得当 $P=Q$ 时, $P \oplus Q = (x^*, y^*)$ 的坐标, 这里从略.

(4) 关于⊕运算的群的性质.

在讨论椭圆曲线上点关于⊕运算构成群的性质之前, 先从下面图中注意观察 $P\oplus Q$ 的意义, 对理解将有帮助.

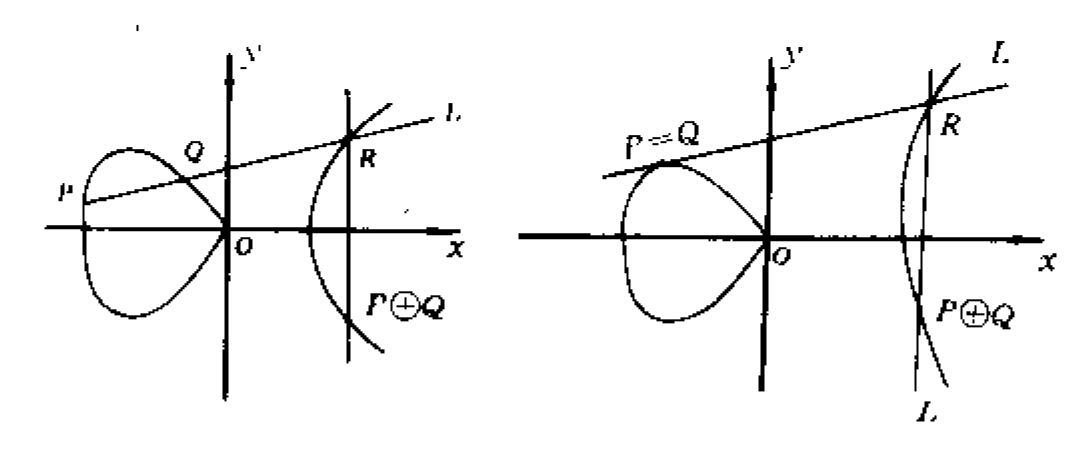


图 2.11.5

定理 2.11.1 若 P 和 Q 是曲线 E 上任意两点，PQ 连线 L 交 E 于另一点 R，则：

(a) $(P \oplus Q) \oplus R = O$

(b) $P \oplus O = P$

(c) $P \oplus Q = Q \oplus P$

(d) E 上存在一点 Q，使得 $P \oplus Q = O$，这样的点 Q 表以 $\ominus P$.

(e) 对于 P 上的任意点 P、Q、R，

$$(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$$

证：(a) $P \oplus Q$ 和 R 作为曲线 E 上两点，它们的连线平行于 y 轴，交 E 于无穷远点 O.

(b) 在曲线 E 上取无穷远点 O 作为 Q 点，PQ 连线平行于 y 轴，交曲线 E 于 R 点. 此时 R 点和 P 点关于 x 轴对称，L 和 L' 重合，所以 $P \oplus O = P$.

(c) 根据 $P \oplus Q$ 的定义，$P \oplus Q = Q \oplus P$ 应成立，即：关于 $\oplus$ 运算交换律成立.

(d) 可通过 P 和 Q 的直线交曲线 E 于 R 点，根据性质 (a) 有：$(P \oplus Q) \oplus R = O$. 根据性质 (b) 有：$P \oplus O = P$，所以，$P \oplus R = O$，$R$ 是所求的 $\ominus P$.

(e) 可通过各种情况对 (e) 进行验证，这过程比较繁琐. 在此从略.

从上面定理可知：若将 O 点看作为运算 $\oplus$ 的零元素，性质 (e) 证明关于 $\oplus$ 运算交换律成立. 性质 (e) 为：若 P 点在曲线 E 上，则曲线 E 上关于 $\oplus$ 结合律成立. 为了方便起见，以后将 $\oplus$ 简化为 $+$，$\ominus$ 记为 $-$. 同样的理由，$P \oplus P$ 记为 $P+P=2P$. 依此类推：

$$mP = \underbrace{P+P+\cdots+P}_{m\text{次}}$$

总之，椭圆曲线上的点关于运算+构成阿贝尔（Abel）群.

例6 对于韦尔斯特拉斯方程

$$y^2 + a_1xy + a_3y - (x^3 + a_2x^2 + a_4x + a_6) = 0$$

定义的曲线 E 上一点 $P(x_1,y_1)$，试证：

$$O - P = (x_1, -y_1 - a_1x - a_3)$$

引 PO 交 E 于 $R(x_1,\overline{y}_1)$，以 $x = x_1$ 代入韦尔斯特拉斯方程得

$$y^2 + a_1x_1y + a_3y - (x_1^3 + a_2x_1^2 + a_4x_1 + a_6) = 0$$

和 $(y - y_1)(y - \overline{y}_1)$ 比较 y 项系数可得：

$$a_1x_1 + a_3 = -y_1 - \overline{y}_1$$

$$\therefore \quad \overline{y}_1 = -y_1 - a_1x_1 - a_3$$

故 $\quad -P = (x_1, -y_1 - a_1x_1 - a_3)$.

例7 $E: y^2=x^3+17$，已知 $P_1=(-2,3)$，$P_2=(2,5)$，求：(a) P_1+P_2；(b) P_1+P_1；(c) $-P_1$.

解：(a) 过点 P_1、P_2 引一直线 L，L 的方程：

$$y - 3 = \frac{5-3}{2-(-2)}(x+2)$$

$$\therefore \quad y=\frac{1}{2}x+4$$

代入 $y^2=x^3+17$，　得：

$$\left(\frac{1}{2}x+4\right)^2=x^3+17$$

$$x^3-\frac{1}{4}x^2-4x+1=0$$

$x_1=2$，　$x_2=-2$　是这方程的两个根.

$$\therefore \quad x^3-\frac{1}{4}x^2-4x+1=(x-2)(x+2)(x-x_3)$$

$\therefore \quad -x_3=-\frac{1}{4}, \quad x_3=\frac{1}{4}$

以之代入 L：$y=\frac{1}{2}x+4$

$$y|_{x=\frac{1}{4}}=\left(\frac{1}{2}x+4\right)_{x=\frac{1}{4}}=\frac{1}{8}+4=\frac{33}{8}$$

$\therefore \quad P_1+P_2=\left(\frac{1}{4}, -\frac{33}{8}\right)$

(b) $y^2=x^3+17$，等式双方对 x 求导得：

$$2y\frac{dy}{dx}=3x^2, \qquad \frac{dy}{dx}=\frac{3x^2}{2y}$$

P 点的切线斜率为：

$$m=\frac{3\cdot(-2)^2}{2\cdot 3}=\frac{12}{6}=2$$

故 P 点的切线方程为

$$y-3=2(x+2)$$

$$y=2x+7$$

$y=2x+7$ 和椭圆曲线 E 的另一交点设为 (x_3,y_3)，则 x_3 满足：

$$(2x+7)^2=x^3+17$$

即 $$4x^2+28x+49=x^3+17$$

$\therefore \quad x^3-4x^2-28x-32=(x^2+4x+4)(x-8)=0$

$\therefore \quad x_3=8, y_3=(2x+7)_{x=8}=23$

故 $$2P_1=(8, -23)$$

(c) $-P_1=(-2, -3)$

(5) 阶的概念.

定义 2.11.1 P 是椭圆曲线 E 上的一个点，若存在最小的正整数 n，使得 $nP=O$，其中 O 是无穷远点，则称 n 是 P 点的阶. 当然，不一定都存在有限的 n，但我们感兴趣的是求椭圆曲线 E 上有有限阶的点，特别是定义在有理数域上的椭圆曲线.

例 8 已知曲线 E 的方程是：$y^2=x^3+1$. 曲线 E 在 $P=$

(2，3) 点的切线斜率为：

$$\left.\frac{dy}{dx}\right|_P=\left.\frac{3x^2}{2y}\right|_{(2,3)}=\frac{12}{6}=2$$

曲线 E 在点 P 的切线方程为：

$$y-3=2(x-2)$$

$$\therefore\quad y=2x-1$$

以 $y=2x-1$ 代入曲线方程，得：

$$(2x-1)^2=x^3+1,\quad 4x^2-4x+1=x^3+1$$

$$\therefore\quad x^3-4x^2+4x=0,x(x-2)^2=0$$

故切线 $y=2x-1$ 交曲线 E 于另一点 $(0,-1)$，或 $2P=(0,1)$. 下面求 $2P+2P$ 点，方法和前面类似.

过 $2P$ 点的切线方程为：

$$y-1=0$$

$$\therefore\quad y=1$$

以 $y=1$ 代入椭圆曲线 E 的方程得 $x=0$. 即切线 $y=1$ 交 E 于 $(0,1)$ 点. 故：

$$4P=(0,-1)$$

过 $4P=(0,-1)$ 和 $2P=(0,1)$ 引连线 L：$x=0$. L 交 E 于另一点 O. 故：

$$6P=O$$

(6) 椭圆曲线的密码系统.

首先叙述一个定理.

定理 2.11.2 如果 E 是定义在域 $GF(q)$ 上的椭圆曲线，N 是 E 上的点 $(x,y)\in GF(q)$ 上的数目. 则

$$|N-(q+1)|\leqslant 2\sqrt{q}$$

该定理由亚汀 (Artin) 提出猜测，后来哈斯 (Hasse) 给了证明. 证明在这里从略.

由前面的讨论可见：椭圆曲线上的点对所定义的加法运算构成 Abel 群.

若我们能找到一椭圆曲线 E，将明文通过编码嵌入到 E 的点，然后在 E 上进行加密. 加密变换也是一种编码. 但从明文到椭圆曲线 E 上的点的编码结果不是密码. 假定嵌入变换已解决，下面讨论在椭圆曲线上的加密，它实际上是将熟知的加密运算移植到椭圆曲线上.

(a) 狄菲—赫尔曼 (Diffie-Hellman) 公钥系统.

已知 q 及 $GF(q)$ 是大家共同确定的. 用户 A 任选一整数 a，$1 \leqslant a \leqslant q-1$，将 a 保密，将 $g^a \in GF(q)$ 公开. g 是 $GF(q)$ 上一固定元素. 用户 B 任选一整数 b，$1 \leqslant b \leqslant q-1$ 予以保密，将 $g^b \in GF(q)$ 公开. 如若 A 和 B 秘密通信，他们之间的通信密钥为 g^{ab}：

$$g^{ab} = (g^a)^b = (g^b)^a$$

而第三者只能知道 g^a 和 g^b 无法得知 g^{ab}，从 g^a 或 g^b 要计算 a 或 b，这是离散对数问题. 是为人熟知的困难问题.

下面介绍如何在椭圆曲线上实现狄菲—赫尔曼的体制，假定椭圆曲线 E 定义在 $GF(q)$ 域上.

设 $P \in E$. 要求由 P 产生的群的元素足够多，P 的作用相当于上述的 g. A 和 B 分别选 a 和 b 予以保密，但将 aP，$bP \in E$ 公开. A、B 间通信用的密钥为 abP，这是第三者无法得知的.

(b) 马塞—阿木拉 (Massey-Omura) 公钥体制.

已知有限域 $GF(q)$，用户 A 选一加密密钥 e_A，$0 \leqslant e_A \leqslant N$，满足 $(e_A, q-1) = 1$，通过欧几里得算法求得 d_A 满足：

$$e_A d_A \equiv (\bmod\ (q-1))$$

类似的理由，若用户 B 有 $e_B d_B \equiv 1 (\bmod\ (q-1))$ 而且 $(e_B, q-1) = 1$，若 A 欲向 B 送去信息 m，则 A 送去 m^{e_A}. B 无法获得

m，因他既不知道 e_A，也不知道 d_A. B 退还 A 以 $m^{e_Ae_B}$. A 对收到的 $m^{e_Ae_B}=c$ 后， 作 $c^{d_A}=m^{e_B}$并送给 B. B 对此结果作 $(m^{e_B})^{d_B}=m^{e_Bd_B}\equiv m(\mathrm{mod}\ q)$，从而获得明文 m.

现在来讨论马塞—阿木拉密码体系在椭圆曲线上的实现.

假定 m 嵌入到 E 上的 P_m 点. 设 E 上的点数 N 为已知的大素数，每一个用户随机选择一数 e，$1<e<N$，$(e,N)=1$. d 是 e 的逆，即 $de\equiv 1(\mathrm{mod}\ N)$. 假如 A 要送 B 信息 m，A 首先送去 e_AP_m，这里 e_A 表示属于用户 A 的 e. B 退还给 A 以 $e_Be_AP_m$. A 再送去 $d_Ae_Be_AP_m=e_BP_m$. B 乘以 d_B 得 $d_Be_BP_m=P_m$，从而获得了 P_m.

(c) 艾格玛尔(E I Gamal)密码系统.

假定在一个有限域 $GF(q)$上讨论. 设 $g\in GF(q)$，g 不为 0 元素. 每一用户随机地选择一数 a，$0<a<q$，a 保密，而将 g^a 公开，如若要向 A 送去信息 m，可随机产生一整数 k，送给 A 以下列一对数：

$$(g^k, mg^{a_Ak})$$

由于 $g^{a_Ak}=(g^{a_A})^k$，所以虽然不知道 a_A，也可以求得 g^{a_Ak}. 但 A 掌握 a_A，他可从 g^k 这个元素得知 g^{a_Ak}，并用 g^{a_Ak} 去“除”第二个数 mg^{a_Ak} 以恢复 m. 这里“除”的含意应理解为用 g^{a_Ak} 的逆元素来乘 mg^{a_Ak}.

同样可在椭圆曲线上实现这个系统，假定明文 m 嵌入到 E 上 P_m 点. 选一点 $B\in E$，它相当于上述的元素 g，每用户都选择一数 a，$0<a<N$，N 是已知数，a 保密，但将 aB 公开. 欲向 A 送 m，可送去下面一对数偶：

$$(kB, P_m+k(a_AB))$$

k 是随机产生的整数. A 可从 kB 求得 ka_AB. 通过：

$$P_m+k(a_AB)-a_AkB=P_m$$

恢复 P_m.

§12 因数分解任斯徒拉(Lenstra)算法

因数分解问题由于密码学的研究而重新被重视. 如若大数的因数分解获成功，则 RSA 等密码系统便宣告失败. 下面介绍椭圆曲线在这方面的应用. 它属于 H · W · 任斯徒拉的工作.

(1) 玻拉 (Pollard) 的 $p-1$ 方法.

玻拉方法假定 n 是合数，p 是 n 的素因子，但 $p-1$ 没有大的因数，方法如下：

S1：求一整数 k，它可被小于某一整数 b 的所有素数除尽，可以取 $k=b!$，也可以取 $k=\text{lcm}\ \{1, 2, \cdots, b\}$

S2：选取整数 a，$2\leqslant a\leqslant n-2$

S3：计算 $a^k \pmod n$.

S4：利用 S3 的结果计算 $\alpha=\gcd\ \{a^k-1, n\}$

S5：如若 α 为 2 或 n 本身，则选新的 a 重新开始.

根据假定 $p-1$ 没有大的因数，可被小于 b 的某正整数除尽. 因而 k 是 $p-1$ 的倍数. 由费尔玛定理：$a^{p-1}\equiv 1 \pmod p$. 所以 $a^k\equiv 1 \pmod p$. 因此 p 可以除尽 $\gcd((a^k-1), n)$. 当 $a^k\equiv 1 \pmod n$ 时方法失败.

例 设 $n=4731$, $k=60$, $a=2$.

$$2^{60} (\text{mod} 4731) \equiv 2^{4\times 3\times 5} (\text{mod}\ 4731)$$
$$\equiv (4096)^5 (\text{mod} 4731)$$
$$\equiv 4096 \cdot (4096^2)^2 (\text{mod}\ 4731)$$

$\because$ $4096^2 \equiv 1090 (\text{mod}\ 4731)$

$\therefore$ $2^{60} (\text{mod}\ 4731) \equiv 4096 \times (1090)^2 (\text{mod}\ 4731)$

$$\equiv 4096 \times 619 (\text{mod } 4731) \equiv 4339 (\text{mod } 4731)$$

又 $4731=4338+393$,

$4338=11\times 393+15$

$393=26\times 15+3$

$15=5\times 3$

$\therefore \quad 4731=3\times 1577$

(2) mod n 导出的椭圆曲线.

定义 2.12.1 假定 p 是 n 的素因子，且 $p>3$，m 是任一整数，若 x_1、x_2 是任意两个有理数，其分母与 m 互素，但 x_1-x_2 化约到最后，分子为被 m 除尽的分数时，写作 $x_1\equiv x_2$ (mod m).

以上是同余式的概念的拓广. 可证若 x_1 是一分母与 m 互素的分数，则存在唯一的整数 x_2，$0\leqslant x_2\leqslant m-1$，使得 $x_1\equiv x_2$ (mod m). 并记 $x_2=(x_1 \text{mod } m)$.

若 $x_1=\dfrac{a}{b}$,$(b,m)=1$,则 $x_1-x_2=\dfrac{a-bx_2}{b}$

$$bx_2\equiv a (\text{mod } m) \tag{2.12.1}$$

上式在 $(b,m)=1$ 时有唯一解 x_2, $0\leqslant x_2\leqslant m-1$.

下面将上述的同余概念用于椭圆曲线 E. 定义一 E(mod m):设 $P(x,y)\in E$,定义

$$P(\text{mod } m)\triangleq (x(\text{mod } m), y(\text{mod } m))$$

则 P(mod m) 为 E(mod m) 椭圆曲线上的点. 前面关于椭圆曲线的讨论均可推广到 E(mod m) 上来，只不过它的每一项都理解为 mod m 意义下的剩余. 比如 $y^2=x^3+ax+b$,其中 a 和 b 都理解为 a(mod m) 和 b(mod m). 特别是 P_1(mod m) + P_2(mod m) 也就是公式(2.11.6). 不过其中每一项都是 mod m 的剩余;分母 (x_1-x_2) 理解为 (x_1-x_2) 的逆,即 $(x_1-x_2)^{-1}$ 使

$(x_1-x_2)\cdot(x_1-x_2)^{-1}\equiv 1(\mathrm{mod}\ m)$. 尤其是$E(\mathrm{mod}\ p)$上的无穷远点$O(\mathrm{mod}\ m)$表示椭圆曲线$E$上坐标的分母有$m$的因子的点.

定理 2.12.1 令$E: y^2=x^3+ax+b$, a, $b\in\mathbf{Z}$, $\gcd\{4a^3+27b^2, n\}=1$. P_1和P_2是E上两点，且$P_2\neq -P_1$. 它们的坐标的分母与n互素，则$P_1+P_2\in E$的坐标的分母与n互素的充要条件是：不存在素数$p|n$，使得曲线$E(\mathrm{mod}\ p)$上$P_1(\mathrm{mod}\ p)$和$P_2(\mathrm{mod}\ p)$之和为$E(\mathrm{mod}\ p)$上的无穷远点$O(\mathrm{mod}\ p)$，其中$E(\mathrm{mod}\ p)$为$GF(p)$域上的椭圆曲线. $O(\mathrm{mod}\ p)$为$GF(p)$域上的椭圆曲线上的无穷远点. 即指的是$(x,y)\in E$上坐标的分母有p因子的点.

证明 必要性. 即已知$P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$, $P_1+P_2=(x_3,y_3)\in E$，它们的坐标的分母都和n互素，p是n的素因子. 则

$$P_1(\mathrm{mod}\ p)+P_2(\mathrm{mod}\ p)\neq O(\mathrm{mod}\ p).$$

分$x_1\not\equiv x_2(\mathrm{mod}\ p)$和$x_1\equiv x_2(\mathrm{mod}\ p)$两种情况进行讨论. 先讨论第一种情况(a)$x_1\not\equiv x_2(\mathrm{mod}\ p)$. 即$P_1\neq P_2$. 根据

$$x_3=\left(\frac{y_2-y_1}{x_2-x_1}\right)^2-x_1-x_2,\quad y_3=-y_1+\left(\frac{y_2-y_1}{x_2-x_1}\right)(x_1-x_3) \tag{2.12.2}$$

显然$P_1(\mathrm{mod}\ p)+P_2(\mathrm{mod}\ p)\neq O(\mathrm{mod}\ p)$成立.

(b)$x_1\equiv x_2(\mathrm{mod}\ p)$，又分$P_1=P_2$和$P_1\neq P_2$两种可能. 若$P_1=P_2$，则$P_1+P_2=2P_1=(x_3,y_3)$，这时

$$x_3=\left(\frac{3x_1^2+a}{2y_1}\right)^2-2x_1,\quad y_3=-y_1+\left(\frac{3x_1^2+a}{2y_1}\right)(x_3-x_1) \tag{2.12.3}$$

其分母$2y_1$不被p除尽. 如若不然，必有$3x_1^2+a$被p除尽，这

说明函数 x^3+ax+b 和它的导函数有相同的零点 $x=x_1$，因 $4a^3+27b^2\not\equiv 0 \pmod p$，这和 mod p 无重根的假定相矛盾. 这也就证明了 $P_1 \pmod p + P_2 \pmod p \neq O \pmod p$. 最后假定 $x_2\equiv x_1 \pmod p$，但 $P_1\neq P_2$，且 $x_2\neq x_1$，令 $x_2=x_1+p^r x$，x 的分母和分子均无 p 的因子，$r\geqslant 1$. 根据 P_1+P_2 的坐标的分母不含 p 的因子的假设.

$$x_3=\left(\frac{y_2-y_1}{x_2-x_1}\right)^2-x_1-x_2,\quad y_3=-y_1+\left(\frac{y_2-y_1}{x_2-x_1}\right)(x_1-x_3)$$

故 $y_2-y_1=p^r y$ 必然要成立，这就证明了

$$y_2\equiv y_1 \pmod p \quad 或 \quad P_1 \pmod p = P_2 \pmod p.$$

另一方面

$$\begin{aligned} y_2^2 &= (x_1+p^r x)^3+a(x_1+p^r x)+b \\ &= x_1^3+3x_1^2 p^r x+3x_1 p^{2r}x^2+p^{3r}x^3+ax_1+ap^r x+b \\ &\equiv x_1^3+3p^r x_1^2 x+ax_1+ap^r x+b \pmod{p^{r+1}} \\ &= x_1^3+ax_1+b+p^r x(3x_1^2+a) \pmod{p^{r+1}} \\ &= y_1^2+p^r x(3x_1^2+a) \pmod{p^{r+1}} \end{aligned} \tag{2.12.4}$$

由于 $P_1\neq P_2$，但 $x_1\equiv x_2 \pmod p$，$y_2\equiv y_1 \pmod p$.

$\therefore\quad P_1 \pmod p + P_2 \pmod p = 2P_1 \pmod p$

$2P_1 \pmod p$ 为 $O \pmod p$ 的充要条件是 $y_1\equiv y_2\equiv 0 \pmod p$，这时从(2.12.4)便有：$(y_2^2-y_1^2)=(y_2-y_1)(y_2-y_1)$ 的分子可被 p^{r+1} 尽除，导致

$$3x_1^2+a\equiv 0 \pmod p$$

这是不可能的. 因多项式 $x^3+ax+b \pmod p$ 没有重根. 这就证明了必要条件.

充分性. 假定 n 的所有素因子 p 有 $P_1 \pmod p - P_2 \pmod p \neq 0 \pmod p$. 证 P_1+P_2 点坐标的分母与 n 互素，也就是它们坐标的分母不被 n 的任意因子 p 除尽.

(1) 若 $x_2 \not\equiv x_1 \pmod p$，根据公式(2.12.2)可知 x_3, y_3 的分母不被 p 除尽，这时定理自然成立.

(2) 若 $x_2 \equiv x_1 \pmod p$，因 $y^2 = x^3 + ax + b$. 故 $y_2 \equiv \pm y_1 \pmod p$，但 $P_1 \pmod p + P_2 \pmod p \neq 0 \pmod p$　故

$$y_2 \equiv y_1 \not\equiv 0 \pmod p$$

在 $x_1 \equiv x_2 \pmod p$，$y_1 \equiv y_2 \pmod p$ 前提下，有两种可能，$P_1 = P_2$ 或 $P_1 \neq P_2$.

若 $P_1 = P_2$，$P_1 + P_2 = 2P_1$，从公式(2.12.3)，这时分母不存在 p 的因子. 若 $P_1 \neq P_2$，令

$$x_2 = x_1 + p^r x,\ r \geqslant 1,\ x \text{ 的分母不含 } p \text{ 的因子}.$$

$$\therefore\quad x_2 - x_1 = p^r x$$

$$\frac{y_2^2 - y_1^2}{x_2 - x_1} \equiv 3x_1^2 + a \pmod p$$

所以 $y_1 + y_2 \equiv 2y_1 \pmod p$

$\dfrac{y_2^2 - y_1^2}{(y_2 + y_1)(x_2 - x_1)} = \dfrac{y_2 - y_1}{x_2 - x_1}$ 的分母不存在 p 的因子. 公式(2.12.2)的分母不存在 p 的因子，也就是说 $P_1 + P_2$ 的坐标的分母不含 p 的因子. 证毕.

(3) 定理 2.12.1 是任斯徒拉因数分解的基础.

给定一合数 n，要找一素数 p，使得 $p|n$，$1 < p < n$.

(a) 首先要随机地生成一椭圆曲线 $E: y^2 = x^3 + ax + b$，及 E 上的点 $P(x, y)$，解法是先随机产生 $P(x, y)$ 点，及整数 a，$b = y^2 - x^3 - ax$. 这样曲线 E 及曲线 E 上的 P 点便产生了. 再检验 $4a^3 + 27b^2 = 0$ 否？这用以保证方程 $x^3 + ax + b = 0$ 没有重根. 若 $4a^3 + 27b^2 = 0$，则改变 a，重复以上的过程.

(b) 给出界 B 和 C，计算

$$k = \prod_{p_i \leqslant B} p_i^{e_i},\quad p_i^{e_i} \leqslant C,\ \forall\ p_i \leqslant B.$$

(c) 计算 $kP(\mathrm{mod}\ n)$，根据公式(2.12.2)，(2.12.3)，x_3 和 y_3 的分母分别为 $x_2 - x_1$ 及 $2y_1$. 求$(x_2 - x_1)$ 和 $2y_1(\mathrm{mod}\ n)$ 的逆时，若发生困难，正好说明分母含有 n 的某一因数. 这正是我们需要的. 转而求它和 n 的最大公因数. 根据定理 2.12.1，$P_1 + P_2$ 的坐标含有 n 的因数 p 的分母的充要条件是

$$P_1(\mathrm{mod}\ p) + P_2(\mathrm{mod} p) = 0(\mathrm{mod}\ p). \quad p|n$$

算法是在 $E(\mathrm{mod}\ n)$ 上进行. 实际上也是在 $E(\mathrm{mod}\ p)$ 上进行，p 是 n 的任意因数.

在计算 kP 的过程中存在一个 k_1，使得 $k_1P(\mathrm{mod}\ p) = 0(\mathrm{mod}\ p)$. 也就是 k_1 为 $P(\mathrm{mod}\ p)$ 点阶的倍数，在计算过程中求 mod n 某分母的逆时，代以求 n 和分母的最大公因数. 这个最大公因数，除非就是 n 本身，一般说来是 n 的因数. 所以在计算 $kP(\mathrm{mod}\ n)$ 时，对某一 $p|n$，k 是 $P(\mathrm{mod}\ p)$ 的阶时，我们将获得一个 n 的真因子.

任斯徒拉方法当所选的 E 和 P 不成功时，换上新的一组重新开始，若失败的概率为 λ，则连续 l 次均失败的概率为 λ^l，故连续 l 次失败的可能性很小.

算法中涉及到选两个整数界 B 和 C，B 和 C 越大当然使得找到 $kP(\mathrm{mod} p) = 0(\mathrm{mod}\ p)$ 的概率增加，然而，计算的时间也越长.

(4) **例 12.2**　已知 $n = 5429$，试通过椭圆曲线因数分解 n.

设令 $B = 3$，　$C = 92$，　取 $E: y^2 = x^3 + 2x - 2$

令 $2P = (\xi_1, \eta_1)$，根据公式(2.12.3)，可得

$$\xi_1 = \left(\frac{3+2}{2}\right)^2 - 2 = \left(\frac{5}{2}\right)^2 - 2 = \frac{17}{4},$$

$$\eta_1 = -1 + \left(\frac{5}{2}\right)\left(1 - \frac{17}{4}\right) = -\frac{73}{8},$$

即　$2P=\left(\frac{17}{4},\ -\frac{73}{8}\right)$，

令　$\frac{17}{4}\equiv u(\bmod 5429)$，　$4u\equiv 17(\bmod 5429)$

$u\equiv 4076(\bmod 5429)$

令　$-\frac{73}{8}\equiv v(\bmod 5429)$，　$8v\equiv -73(\bmod 5429)$

$v\equiv 3384(\bmod 5429)$

故　$2P(\bmod 5429)=(4076,3384)$.

当然也可以用下面的步骤计算 $2P(\bmod 5429)$.

$$u\equiv\left(\frac{3+2}{2}\right)^2-2\equiv\left(\frac{5}{2}\right)^2-2(\bmod 5429)$$

但 $\frac{1}{2}$ 理解为 $2^{-1}(\bmod 5429)$，即上面的计算在 $E(\bmod 5429)$ 上进行. $2^{-1}(\bmod 5429)\equiv 2715$

$$\begin{aligned}\therefore\quad u&\equiv(2715\times 5)^2-2\equiv(2717)^2-2\\&\equiv 4078(\bmod 5429)\end{aligned}$$

同样

$$\begin{aligned}v&\equiv-1+\left(\frac{5}{2}\right)\left(\frac{-13}{4}\right)\\&\equiv-1+(2715\times 5)(-13\times 4072)\\&\equiv-1+(2717)(1354)(\bmod 5429)\\&\equiv 3384(\bmod 5429)\end{aligned}$$

即 $2P(\bmod 5429)=(4076,3384)$，结果是一样，这里 $4^{-1}(\bmod 5429)=4072$.

依类似的算法计算 $4P(\bmod 5429)$. 令 $2^2P(\bmod 5429)=(u_2,v_2)$，下面计算假定在 $GF(p)$ 上进行，其中 $P|5429$.

$$\begin{aligned}u_2&\equiv\left(\frac{3\times 4076+2}{6768}\right)^2-8152(\bmod 5429)\\&\equiv\left(\frac{1549+2}{1339}\right)^2-2723(\bmod 5429)\end{aligned}$$

$1339^{-1} \equiv 2826 \pmod{2723}$.

$\therefore \quad u_2 \equiv (1551 \times 2826)^2 - 2723 \pmod{5429}$

$\equiv (1923)^2 - 2723 \pmod{5429}$

$\equiv 780 - 2723 \pmod{5429}$

$\equiv -1943 \pmod{5429} \equiv 3486 \pmod{5429}$

$v_2 \equiv -3384 + (1551 \times 2826)(4076 - 3486) \pmod{5429}$

$\equiv -3384 + 1923 \times 590 \pmod{5429}$

$\equiv -3384 + 5338 \pmod{5429}$

$\equiv 1954 \pmod{5429}$.

$\therefore \quad 4P \pmod{5429} = (3486, 1954)$

类似的步骤计算 $2^{\alpha}3^{\beta}P \pmod{5429}$.

当 $\alpha=6$，$\beta=2$ 时，发现分母与 5429 有 61 的公因子.

$5429=61\times 89$

(5) 附带说明取 $c=92$ 的依据：$n=5429$，$\sqrt{5429}=73.68$，即 n 的因数 $p<\sqrt{n}=73.68$，定义在 $GF(p)$上的椭圆曲线 E 上的属于 $GF(p)$的点的数目 N 应满足

$$N\leqslant p+1+2\sqrt{p}=74+2\sqrt{73}$$
$$<74+2\sqrt{81}=92$$

取 $c=92$ 盖出于 $c>N$ 的考虑.

§13 编码理论简介

在近代密码学研究中有一种努力，企图将纠错码用在密码上，由于纠错码有着许多独特的功能，所以它具有潜在的优势. 首先可一次编码完成对信息的加密和纠错两种功能. 提高冗余度码元的双重作用，在保密与纠错间找到一种优美的折衷. 因

为一般说来两者是有矛盾的.

(1) 为了讨论方便起见，有必要介绍纠错码理论，对已掌握这部分内容的读者可免去.

通常 (n, l) 组码表示由 l 位 0，1 字符串到 n 位 0，1 字符串的编码过程，即编码的过程为将明文

$$m=m_1m_2\cdots m_l \qquad m_i\in\{0, 1\}, i=1, 2, \cdots, l$$

变换为码文

$$E(m)=w=w_1w_2\cdots w_n \qquad w_i\in\{0, 1\}$$
$$i=1, 2, \cdots, n$$

码文在信道中传输由于受到干扰难免出错，编码的目的在于发现错，这是检错. 进而将出错处纠正过来，这是纠错. 所以检错和纠错是编码理论所要研究的内容，当然密码也是一种编码，它则是以保密和安全为目的. 最简单的有：

例 1 $(n+1, n)$ 检错码，即在 n 位明文后面附加一位奇偶校验位，即

$$E(m)=m_1m_2\cdots m_nm_{n+1}$$

如果传输过程中出一个错，即将某一位的 0 错为 1，或将 1 错为 0，则可立即被发现. 当然在 n 位中若出两个错，则反而发现不了，当然在 n 位中出两个错的可能性比较小. 这样的码并不能判定是哪位出了差错，所以只是检错码.

例 2 $(3n, n)$ 码，即

$$E(m)=m_1m_2\cdots m_nm_1m_2\cdots m_nm_1m_2\cdots m_n$$

即对每组 n 位的码连续传输 3 遍，若某一位出现不同，则必有两组是一样的，便以出现两次一样的为准，达到纠错的目的. 当然，同时在同一位上出两次或三次错的可能性虽存在，但终究是不多见. 某一位三次传输相同，由于错的一样而被认为正确的，而出错的概率是很小的. 例如出错率为 0.001，传输正确的

概率为 0.999，三次都正确的概率为 $(0.999)^3=0.997003$，出现 1 个错的概率为

$$\binom{3}{1}(0.001)(0.999)^2=0.002994$$

所以 $(3n, n)$ 码传输正确的概率包含 3 次都正确及最多出一次错的概率之和．故为

$$0.997003+0.002994=0.999997$$

所以传输正确的概率从每位为 0.999，提高到 0.999997，但码文比明文扩大了 3 倍，这很难为人们所接受．

(2) 令 $m=m_1m_2\cdots m_n$，$m'=m_1'm_2'\cdots m_n'$ 都是长度为 n 的 0，1 符号串，$w(m)$ 为 m 的 n 位中 1 的数目，称为 m 的权．

$d(m,m')=w(m\oplus m')$ 称为 m 和 m' 间的距离．

由于 $1\oplus 1=0$，$0\oplus 0=0$，$1\oplus 0=0\oplus 1=1$，所以 $d(m,m')$ 实际上是 m 和 m' 不同的位的数目．以后 $\oplus$ 简记为 $+$，不必特别声明．

引理 2.13.1 若 $\boldsymbol{a}$，$\boldsymbol{b}$，$\boldsymbol{c}$ 都是长度为 n 的 0，1 符号串，即

$\boldsymbol{a}=a_1a_2\cdots a_n$

$\boldsymbol{b}=b_1b_2\cdots b_n$

$\boldsymbol{c}=c_1c_2\cdots c_n$

a_i、b_i、$c_i\in\{0, 1\}$，$i=1, 2, \cdots, n$．则

(a) $d(\boldsymbol{a},\boldsymbol{b})=d(\boldsymbol{b},\boldsymbol{a})$，

(b) $d(\boldsymbol{a},\boldsymbol{c})\leqslant d(\boldsymbol{a},\boldsymbol{b})+d(\boldsymbol{b},\boldsymbol{c})$．

证明：

(a) $d(\boldsymbol{a},\boldsymbol{b})=w(\boldsymbol{a}+\boldsymbol{b})=w(\boldsymbol{b}+\boldsymbol{a})$

$\qquad\qquad=d(\boldsymbol{b},\boldsymbol{a})$．

(b) 若 $a_i\neq c_i$，则有两种可能，即

$a_i=b_i$ 但 $b_i\neq c_i$，或 $a_i\neq b_i$ 但 $b_i=c_i$

当然反过来若 $a_i=c_i$ 时，也有两种可能，一种是 $a_i=b_i=c_i$，另一种则是 $a_i\neq b_i$，$b_i\neq c_i$，基于以上的讨论，令

$$d(a_i,b_i)=\begin{cases}0, & a_i=b_i\\ 1, & a_i\neq b_i\end{cases}$$

$$\begin{aligned} d(\boldsymbol{a},\boldsymbol{c}) &= \sum_{i=1}^{n} d(a_i,c_i)\\ &\leqslant \sum_{i=1}^{n} d(a_i,b_i)+\sum_{i=1}^{n} d(b_i,c_i)\\ &\leqslant d(\boldsymbol{a},\boldsymbol{b})+d(\boldsymbol{b},\boldsymbol{c}). \end{aligned}$$

定理 2.13.1 一组码可以检出 k 个错误的充要条件是码间最短距离至少为 $k+1$.

证明：设 $\boldsymbol{w}=w_1w_2\cdots w_n$ 是码字，$\boldsymbol{w}$ 传输中有误差 $\boldsymbol{e}$，指收到的是 $\boldsymbol{r}=\boldsymbol{w}+\boldsymbol{e}$，$\boldsymbol{e}$ 能被检出的充要条件是 $\boldsymbol{w}+\boldsymbol{e}$ 不是码字. 因此能检出 k 个错误的充要条件是码句之间的距离至少为 $k+1$.

定理 2.13.2 若两个码字之间的最短距离为 $2k+1$，则可以纠正权不超过 k 的误差.

证：设码字 $\boldsymbol{a}$ 传输过程发生误差，得到的是 $\boldsymbol{r}$，且 $d(\boldsymbol{a},\boldsymbol{r})\leqslant k$，则不存在别的码字与 $\boldsymbol{r}$ 的距离不超过 k. 如若不然，设为 $\boldsymbol{b}$，满足 $d(\boldsymbol{r},\boldsymbol{b})\leqslant k$，将有

$$\begin{aligned} d(\boldsymbol{a},\boldsymbol{b}) &\leqslant d(\boldsymbol{a},\boldsymbol{r})+d(\boldsymbol{r},\boldsymbol{b})\\ &\leqslant k+k<2k+1. \end{aligned}$$

与假设矛盾.

定理得证.

定理 2.13.2 是极大似然译码方法的依据，例如有码字

10010　01001　10101　01110

两两作 $\oplus$ 运算如下：

	10010	01001	10101	01110
10010	——	11011	00111	11100
01001	11011	——	11100	00111
10101	00111	11100	——	11011
01110	11100	00111	11011	——

$\min\limits_{i\neq j}\{d(a_i, a_j)\}=3$，故能纠正一个错误.

w	10010	01001	10101	01110
r	00010	11001	00101	11110
	11010	00001	11101	00110
	10110	01101	10001	01010
	10000	01011	10111	01100
	10011	01000	10100	01111

上表给出距离为 1 的译码表，比如 $r=01011$，它与 01001 距离为 1，故译为 01001. 认为第 4 位出错. 码字数量大时靠译码表译码工作量和存储量都太大.

(3) 生成矩阵.

令

$$G=(g_{ij})_{l\times n}=\begin{pmatrix}G_1\\G_2\\\vdots\\\vdots\\G_l\end{pmatrix}=\begin{pmatrix}g_{11}&g_{12}&\cdots&g_{1n}\\g_{21}&g_{22}&\cdots&g_{2n}\\\cdots&\cdots&&\cdots\\g_{l1}&g_{l2}&\cdots&g_{ln}\end{pmatrix}$$

其中 G_i 为 G 矩阵的第 i 行向量，即

$$G_i = (g_{i1}g_{i2}\cdots g_{in})$$

$g_{ij} \in \{0,1\}, i = 1,2,\cdots,l; j = 1,2,\cdots,n.$

设 $m = m_1m_2\cdots m_l$ 为信息块，或长度为 l 的明文块，编码过程为

$$E(m) = (m_1m_2\cdots m_l)\begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \cdots & \cdots & & \cdots \\ g_{l1} & g_{l2} & \cdots & g_{ln} \end{pmatrix} = mG$$

$$= \sum_{i=1}^{l} m_i G_i = W$$

例如

$G = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$ 对于信息 1 0 1，对应的码字为

$$W = (1\ 0\ 1)\begin{pmatrix} 1\ 0\ 0\ 1\ 1\ 0 \\ 0\ 1\ 0\ 0\ 1\ 1 \\ 0\ 0\ 1\ 1\ 0\ 1 \end{pmatrix}$$

$$= (1\ 0\ 0\ 1\ 1\ 0) + (0\ 0\ 1\ 1\ 0\ 1)$$

$$= (1\ 0\ 1\ 0\ 1\ 1)$$

现将长度为 3 的信息和对应的码字列表于下：

信　息	码　　字
0 0 0	0 0 0 0 0 0
0 0 1	0 0 1 1 0 1
0 1 0	0 1 0 0 1 1
0 1 1	0 1 1 1 1 0
1 0 0	1 0 0 1 1 0
1 0 1	1 0 1 0 1 1
1 1 0	1 1 0 1 0 1
1 1 1	1 1 1 0 0 0

显然，码字（6 位）前 3 位为信息位，余下的 3 位是校验位．可将这种情况推广到（n，l）码上．即

$$G=\begin{pmatrix}1 & 0 & 0 & \cdots & 0 & g_{11} & g_{12} & \cdots & g_{1\,n-l} \\ 0 & 1 & 0 & \cdots & 0 & g_{21} & g_{22} & \cdots & g_{2\,n-l} \\ & & & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & g_{l1} & g_{l2} & \cdots & g_{l\,n-l}\end{pmatrix}_{l\times n}$$

上，信息 $m=m_1m_2\cdots m_l$ 经编码过程得 n 位码字

信息位	校验位
l 位	$n-l$ 位

为了方便起见，假定明文 $m=m_1m_2\cdots m_l$ 对应的码文是

$$W=(m_1m_2\cdots m_l)\begin{pmatrix}1 & 0 & 0 & \cdots & 0 & g_{11} & g_{12} & \cdots & g_{1\,n-l} \\ 0 & 1 & 0 & \cdots & 0 & g_{21} & g_{22} & \cdots & g_{2\,n-l} \\ & & & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & g_{l1} & g_{l2} & \cdots & g_{l\,n-l}\end{pmatrix}_{l\times n}$$

$$=(m_1m_2m_3\cdots m_lm_{l+1}\cdots m_n)=r$$

显然有

$$m_{l+j}=\sum_{i=1}^{l}g_{ij}m_i,\qquad j=1,2,\cdots,n-l \tag{2.13.1}$$

或

$$g_{1j}m_1+g_{2j}m_2+\cdots+g_{lj}m_l+m_{l+j}=0$$

$$j=1,\ 2,\ \cdots,\ n-l$$

也就是说 m_{l+1}，m_{l+2}，$\cdots$，m_n 即附加的多余部份是用来检验传输过程是否出错及哪些出错？所以叫做校验位．$n-l$ 个方程（2.13.1）可以用矩阵形式表示为

$$\begin{pmatrix} g_{11} & g_{21} & \cdots & g_{l1} & 1 & 0 & \cdots & 0 \\ g_{12} & g_{22} & \cdots & g_{l2} & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & & & \ddots & \\ g_{1,n-l} & g_{2,n-l} & \cdots & g_{l,n-l} & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

或写作

$$Hr^{\mathrm{T}}=O \qquad (2.13.2)$$

称之为校验方程，$H=(h_{ij})_{(n-l)\times n}$ 称为校验矩阵. 显然生成矩阵 G 和校验矩阵 H 有如下关系. 若

$$G=(E_{(l)} \vdots A)_{l\times n}$$

其中 $A=(a_{ij})_{l\times(n-l)}$，$E_{(l)}$ 为 l 阶单位矩阵，则有

$$H=(A^{\mathrm{T}} \vdots E_{(n-l)})_{(n-l)\times n}$$

校验矩阵可用来纠正错误.（2.13.2）说明正确传输应满足的等式. 如若 $r=W+e$，e 是误差.

$$Hr^{\mathrm{T}}=H(W+e)^{\mathrm{T}}=HW^{\mathrm{T}}+He^{\mathrm{T}}=He^{\mathrm{T}}.$$

若 $He^{\mathrm{T}}\neq 0$，可由 He^{T} 看出究竟第几位出了错给予纠正. 以

$$H=\begin{pmatrix} 1&0&1&1&0&0 \\ 1&1&0&0&1&0 \\ 0&1&1&0&0&1 \end{pmatrix} \qquad r=1\ 0\ 0\ 1\ 0\ 1$$

为例

$$Hr^{\mathrm{T}}=\begin{pmatrix} 1&0&1&1&0&0 \\ 1&1&0&0&1&0 \\ 0&1&1&0&0&1 \end{pmatrix}\begin{pmatrix} 1\\0\\0\\1\\0\\1 \end{pmatrix}=\begin{pmatrix} 0\\1\\1 \end{pmatrix}$$

$(0\ 1\ 1)^{\mathrm{T}}$ 正好是 H 矩阵的第 2 列，故可知第 2 位出错. 将 $r=1\ 0\ 0\ 1\ 0\ 1$ 的第 2 位的 0 改为 1，故得

$W=1\,1\,0\,1\,0\,1$

信息位为 1 1 0，译码便结束.

当然，若出现两个以上的错误，这种方法便失败，也就是说不能获得正确的纠错，现在将译码步骤列下. 设收到的信息为

$r=r_1r_2\cdots r_n$

S1：计算 $S=Hr^t$；

S2：若 $S=O$，则可认为传输过程是正确的，则明文 $m=r_1r_2\cdots r_l$. 若 $S\neq O$ 转 S3；

S3：若 S 是矩阵 H 的第 i 列，则认为 r_i 有错误，予以改正. 然后取前面的 l 位作为明文；若 S 不是 H 的列向量（且不为零），则认为传输过程至少出现两个以上的错误，无法正确纠错. 以上的 S 通常叫校正子.

定理 2.1.3.3 $(n-l)\times n$ 的校验矩阵能正确纠正 1 个错误的充要条件是 H 的各列为不相同的非零列向量.

这个定理很显然，证明由读者自己来完成.

(4) 依据 $k\times n$ 阶矩阵 H 能纠一个错的充要条件是 H 矩阵的各列为不相同的非零向量，当 k 确定后，n 最大可取 2^k-1. 又因 $k=n-l$，故 $l=n-k$.

例如，$k=3$ 时，$n=2^3-1=7$，$l=7-3=4$

例如，

$$H=\begin{pmatrix}0&1&1&1&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{pmatrix}$$

$\boldsymbol{H}$ 的各列包含了除 $\begin{pmatrix}0\\0\\0\end{pmatrix}$ 以外的所有状态，对应的生成矩阵为

$$
\boldsymbol{G}=\begin{pmatrix}
1&0&0&0&0&1&1\\
0&1&0&0&1&0&1\\
0&0&1&0&1&1&0\\
0&0&0&1&1&1&1
\end{pmatrix}
$$

用这种办法构成的可纠一个错的码称为（7，4）汉明（Hamming）码.

又如　$k=4$ 时，$n=2^4-1=15$，$l=n-k=15-4=11$

也可以构造汉明码，0，1 字符串长度为 11 的明文经编码得长度为 15 的码字.

下面便是（15，11）的汉明码的校验矩阵.

$$
\boldsymbol{H}=\begin{pmatrix}
1&1&1&0&0&0&1&1&1&0&1&1&0&0&0\\
1&0&0&1&1&0&1&1&0&1&1&0&1&0&0\\
0&1&0&1&0&1&1&0&1&1&1&0&0&1&0\\
0&0&1&0&1&1&0&1&1&1&1&0&0&0&1
\end{pmatrix}
$$

从校验矩阵不难求得对应的生成矩阵 $\boldsymbol{G}$，这里从略.

§14　BCH 码和郭帕(Goppa)码

(1) BCH 是 Bose，Chaudhari，Hocquenghem 的缩写. BCH 码是他们三人于 1959 年同时发明的. 现在假定读者对有限域 $GF(p^n)$有基本的了解.

假定 α 是域 $GF(2^4)$ 的本元元素，满足 $\alpha^4+\alpha+1=0$. 即 $\bmod(\alpha^4+\alpha+1)$ 构成一个 $GF(2^4)$ 域. 令

$$H=\begin{pmatrix}1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{14}\\ 1 & \alpha^3 & (\alpha^2)^3 & (\alpha^3)^3 & \cdots & (\alpha^{14})^3\end{pmatrix} \qquad (2.14.1)$$

由于 $\alpha^0=1$，$\alpha^1=\alpha$，α^2，α^3，$\alpha^4=1+\alpha$，$\alpha^5=\alpha+\alpha^2$，$\alpha^6=\alpha^2+\alpha^3$，$\alpha^7=\alpha^3+\alpha^4=1+\alpha+\alpha^3$，$\alpha^8=\alpha+\alpha^2+\alpha^4=1+\alpha^2$，$\alpha^9=\alpha+\alpha^3$，$\alpha^{10}=\alpha^2+\alpha^4=1+\alpha+\alpha^2$，$\alpha^{11}=\alpha+\alpha^2+\alpha^3$，$\alpha^{12}=\alpha^2+\alpha^3+\alpha^4=1+\alpha^2+\alpha^3+\alpha$，$\alpha^{13}=\alpha+\alpha^2+\alpha^3+\alpha^4=1+\alpha^2+\alpha^3$，$\alpha^{14}=\alpha+\alpha^3+\alpha^4=1+\alpha^3$，$\alpha^{15}=\alpha+\alpha^4=1$.

$$\therefore\quad H=\begin{pmatrix}1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \cdots & \alpha^{14}\\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \cdots & \alpha^{12}\end{pmatrix}$$

用下列矩阵来表示

$$H=\begin{pmatrix}
1&0&0&0&1&0&0&1&1&0&1&0&1&1&1\\
0&1&0&0&1&1&0&1&0&1&1&1&1&0&0\\
0&0&1&0&0&1&1&0&1&0&1&1&1&1&0\\
0&0&0&1&0&0&1&1&0&1&0&1&1&1&1\\
 & & & &\cdots\cdots & & & & &\cdots\cdots & & & & & \\
1&0&0&0&1&1&0&0&0&1&1&0&0&0&1\\
0&0&0&1&1&0&0&0&1&1&0&0&0&1&1\\
0&0&1&0&1&0&0&1&0&1&0&0&1&0&1\\
0&1&1&1&1&0&1&1&1&1&0&1&1&1&1
\end{pmatrix} \qquad (2.14.2)$$

这里用

$$\begin{pmatrix}0\\1\\0\\1\end{pmatrix},\quad \begin{pmatrix}1\\0\\1\\1\end{pmatrix}$$

分别表示 $\alpha+\alpha^3=\alpha^9$，$1+\alpha^2+\alpha^3=\alpha^{13}$. 其它以此类推.

若在第 h，k 两位发生错误，则校验子

$$S=\begin{pmatrix}\alpha^h+\alpha^k\\ \alpha^{3h}+\alpha^{3k}\end{pmatrix}=\begin{pmatrix}z_1\\ z_2\end{pmatrix}$$

$$z_1=\alpha^h+\alpha^k,\quad z_2=\alpha^{3h}+\alpha^{3k}$$

$$\therefore\quad z_2=(\alpha^h+\alpha^k)(\alpha^{2h}+\alpha^{h+k}+\alpha^{2k})$$
$$=z_1(z_1^2+\alpha^{h+k})$$

$$\alpha^{h+k}=\frac{z_2}{z_1}+z_1^2$$

故 α^h 和 α^k 是方程

$$z^2+z_1z+\left(\frac{z_2}{z_1}+z_1^2\right)=0 \qquad (2.14.3)$$

的两个根.

若 $z_1=z_2=0$，则认为无差错，若 $z_2=z_1^3=\alpha^{2h}$，则认为有一个错误发生在第 h 位.

例 1 若在第 7、9 位发生错误，则

$$z_1=\begin{pmatrix}0\\0\\1\\1\end{pmatrix}+\begin{pmatrix}1\\0\\1\\0\end{pmatrix}=\begin{pmatrix}1\\0\\0\\1\end{pmatrix}=\alpha^{14}$$

$$z_2=\begin{pmatrix}0\\0\\0\\1\end{pmatrix}+\begin{pmatrix}0\\1\\0\\1\end{pmatrix}=\begin{pmatrix}0\\1\\0\\0\end{pmatrix}=\alpha=\alpha^{16}$$

$$\frac{z_2}{z_1}+z_1^2=\alpha^3+\alpha^{13}=\alpha^3+1+\alpha^2+\alpha^3=1+\alpha^3=\alpha^{14}$$

$$\therefore\quad z^2+\alpha^{14}z+\alpha^{14}=(z+\alpha^6)(z+\alpha^8)=0$$

这就检证了 BCH 纠错是正确的.

(2) 长为 n 的郭帕码要用到两个概念，一是系数在 $GF(p^m)$ 的郭帕多项式 $G(z)$；另一个是 $GF(p^m)$ 满足条件 $G(\beta_i)\neq 0, i=1,2,\cdots,n$ 的子集 $L=\{\beta_1,\beta_2,\cdots,\beta_n\}$.

关于 $GF(p)$ 的任一向量 $\boldsymbol{a}=\{a_1,\ a_2,\ \cdots,\ a_n\}$，有一有理函数

$$R_a(z) = \sum_{i=1}^{n} \frac{a_i}{z-\beta_i} \tag{2.14.4}$$

由满足

$$R_a(z) \equiv 0(\mathrm{mod}G(z))$$

的所有向量 a 构成的码称为郭帕码，用 Γ（L，G）表示之．郭帕码的校验矩阵不难求得．

$$\because \quad -(z-\beta_i)\frac{(G(z)-G(\beta_i))}{z-\beta_i}G(\beta_i)^{-1}$$

$$=-G(z)G(\beta_i)^{-1}+G(\beta_i)G(\beta_i)^{-1}$$

$$\equiv 1(\mathrm{mod}\ G(z))$$

由此可得 mod $G(z)$，$(z-\beta_i)$的逆，即

$$(z-\beta_i)^{-1}=-\frac{G(z)-G(\beta_i)}{z-\beta_i}G(\beta_i)^{-1}$$

所以 $a \in \Gamma(L,G)$ 的充要条件是

$$\sum_{i=1}^{n} a_i \frac{G(z)-G(\beta_i)}{z-\beta_i}G(\beta_i)^{-1}=0 \tag{2.14.5}$$

若 $G(z)=\sum_{i=0}^{r} g_i z^i$， $g_i \in GF(p^m)$， $g_r \neq 0$，则由于

$$(z-\beta_i)(z^{k-1}+z^{k-2}\beta_i+\cdots+\beta_i^{k-1})=z^k-\beta_i^k,$$

$$k=2,3,\cdots,r$$

$$(z-\beta_i)[g_r(z^{r-1}+z^{r-2}\beta_i+\cdots+\beta_i^{r-1})$$
$$+g_{r-1}(z^{r-2}+z^{r-3}\beta_i+\cdots+\beta_i^{r-2})+\cdots$$
$$+g_2(z+\beta_i)+g_1]$$
$$=g_r(z-\beta_i)(z^{r-1}+z^{r-2}\beta_i+\cdots+\beta_i^{r-1})$$
$$+g_{r-1}(z-\beta_i)(z^{r-2}+z^{r-3}\beta_i+\cdots+\beta_i^{r-2})$$
$$+\cdots+g_2(z-\beta_i)(z+\beta_i)+g_1(z-\beta_i)$$
$$=g_r(z^r-\beta_i^r)+g_{r-1}(z^{r-1}-\beta_i^{r-1})+\cdots$$
$$+g_2(z^2-\beta_i^2)+g_1(z-\beta_i)$$

$$= G(z) - G(\beta_i)$$

这就证明了

$$\frac{G(z) - G(\beta_i)}{z - \beta_i} = g_r(z^{r-1} + z^{r-2}\beta_i + \cdots + \beta_i^{r-1}) + g_{r-1}(z^{r-2} + z^{r-3}\beta_i + \cdots + \beta_i^{r-2}) + \cdots + g_2(z + \beta_i) + g_1$$

根据 $a \in \Gamma(L,G)$ 的充要条件(2.14.5)，

$$\therefore \quad \sum_{i=1}^{n} a_i \frac{G(z) - G(\beta_i)}{z - \beta_i} G(\beta_i)^{-1}$$

$$= \sum_{i=1}^{n} a_i [g_r(z^{r-1} + z^{r-2}\beta_i + \cdots + \beta_i^{r-1}) + g_{r-1}(z^{r-2} + z^{r-3}\beta_i + \cdots + \beta_i^{r-2}) + \cdots + g_2(z + \beta_i) + g_1] G(\beta_i)^{-1}$$

$$= z^{r-1} g_r \left[\sum_{i=1}^{n} a_i G(\beta_i)^{-1}\right] + z^{r-2}\left[\sum_{i=1}^{n} a_i (g_r\beta_i + g_{r-1}) G(\beta_i)^{-1}\right] + z^{r-3}\left[\sum_{i=1}^{n} a_i (g_r\beta_i^2 + g_{r-1}\beta_i + g_{r-2}) G(\beta_i)^{-1}\right] + \cdots + z\sum_{i=1}^{n} a_i (g_r\beta_i^{r-2} + g_{r-1}\beta_i^{r-3} + \cdots + g_2) G(\beta_i)^{-1} + \sum_{i=1}^{n} a_i (g_r\beta_i^{r-1} + g_{r-1}\beta_i^{r-2} + \cdots + g_1) G(\beta_i)^{-1} = 0$$

令 z^{r-1}，z^{r-2}，…，r，1 的系数分别等于 0，故有

$$a_1 g_r G(\beta_1)^{-1} + a_2 g_r G(\beta_2)^{-1} + \cdots + a_n g_r G(\beta_n)^{-1} = 0$$

$$a_1(\beta_1 g_r + g_{r-1}) G(\beta_1)^{-1} + a_2(\beta_2 g_r + g_{r-1}) G(\beta_2)^{-1} + \cdots + a_n(\beta_n g_r + g_{r-1}) G(\beta_n)^{-1} = 0$$

$$a_1(\beta_1^2 g_r + \beta_1 g_{r-1} + g_{r-2}) G(\beta_1)^{-1} + a_2(\beta_2^2 g_r + \beta_2 g_{r-1})$$

$$+ g_{r-2})G(\beta_2)^{-1} + \cdots + a_n(\beta_n^2 g_r + \beta_n g_{r-1} + g_{r-2})G(\beta_n)^{-1} = 0$$

……

$$a_1(\beta_1^{r-1} g_r + \beta_1^{r-2} g_{r-1} + \cdots + g_1)G(\beta_1)^{-1} + a_2(\beta_2^{r-1} g_r + \beta_2^{r-2} g_{r-1} + \cdots + g_1)G(\beta_2)^{-1} + \cdots + a_n(\beta_n^{r-1} g_r + \beta_n^{r-2} g_{r-1} + \cdots + g_1)G(\beta_n)^{-1} = 0$$

写成矩阵形式：

$$\begin{pmatrix} g_r G(\beta_1)^{-1} & \cdots & g_r G(\beta_n)^{-1} \\ (\beta_1 g_r + g_{r-1}) G(\beta_1)^{-1} & \cdots & (\beta_n g_r + g_{r-1}) G(\beta_n)^{-1} \\ (\beta_1^2 g_r + \beta_1 g_{r-1} + g_{r-2}) G(\beta_1)^{-1} & \cdots & (\beta_n^2 g_r + \beta_n g_{r-1} + g_{r-2}) G(\beta_n)^{-1} \\ \cdots & & \\ (\beta_1^{r-1} g_r + \beta_1^{r-2} g_{r-1} + \cdots + g_1) G(\beta_1)^{-1} & \cdots & (\beta_n^{r-1} g_r + \beta_n^{r-2} g_{r-1} + \cdots + g_1) G(\beta_n)^{-1} \end{pmatrix}$$

$$\times \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{pmatrix} = 0$$

或写作

$$\boldsymbol{H}\boldsymbol{a}^T = 0$$

称 H 为校验矩阵，而且有

$$\begin{pmatrix} g_r G(\beta_1)^{-1} & \cdots & g_r G(\beta_n)^{-1} \\ (\beta_1 g_r + g_{r-1}) G(\beta_1)^{-1} & \cdots & (\beta_n g_r + g_{r-1}) G(\beta_n)^{-1} \\ (\beta_1^2 g_r + \beta_1 g_{r-1} + g_{r-2}) G(\beta_1)^{-1} & \cdots & (\beta_n^2 g_r + \beta_n g_{r-1} + g_{r-2}) G(\beta_n)^{-1} \\ \cdots & & \\ (\beta_1^{r-1} g_r + \beta_1^{r-2} g_{r-1} + \cdots + g_1) G(\beta_1)^{-1} & \cdots & (\beta_n^{r-1} g_r + \beta_n^{r-2} g_{r-1} + \cdots + g_1) G(\beta_n)^{-1} \end{pmatrix}$$

$$
=\begin{pmatrix} g_r & 0 & 0 & \cdots & 0 \\ g_{r-1} & g_r & 0 & \cdots & 0 \\ g_{r-2} & g_{r-1} & g_r & \cdots & 0 \\ & & & \ddots & \\ g_1 & g_2 & g_3 & g_r & \end{pmatrix}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \beta_1 & \beta_2 & \cdots & \beta_n \\ \beta_1^2 & \beta_2^2 & \cdots & \beta_n^2 \\ & & & \\ \beta_1^{r-1} & \beta_2^{r-1} & \cdots & \beta_n^{r-1} \end{pmatrix}
$$

$$
\times\begin{pmatrix} G(\beta_1)^{-1} & & & & \\ & G(\beta_2)^{-1} & & 0 & \\ & & G(\beta_3)^{-1} & & \\ 0 & & & \ddots & \\ & & & & G(\beta_n)^{-1} \end{pmatrix}=\boldsymbol{LAR}
$$

其中

$$
\boldsymbol{L}=\begin{pmatrix} g_r & 0 & 0 & \cdots & 0 \\ g_{r-1} & g_r & 0 & \cdots & 0 \\ g_{r-2} & g_{r-1} & g_r & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \\ g_1 & g_2 & g_3 & \cdots & g_r \end{pmatrix}
$$

$$
\boldsymbol{A}=\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_n \\ \beta_1^2 & \beta_2^2 & \beta_3^2 & \cdots & \beta_n^2 \\ \cdots & \cdots & \cdots & & \\ \beta_1^{r-1} & \beta_2^{r-1} & \beta_3^{r-1} & \cdots & \beta_n^{r-1} \end{pmatrix}
$$

$$
\boldsymbol{R}=\begin{pmatrix} G(\beta_1)^{-1} & & & \\ & G(\beta_2)^{-1} & 0 & \\ 0 & & \ddots & \\ & & & G(\beta_n)^{-1} \end{pmatrix}
$$

由于 $\boldsymbol{L}$ 是可逆矩阵，故可取 $\boldsymbol{AR}$ 作为校验矩阵.

$$H' = AR\begin{bmatrix} G(\beta_1)^{-1} & G(\beta_2)^{-1} & \cdots & G(\beta_n)^{-1} \\ \beta_1 G(\beta_1)^{-1} & \beta_2 G(\beta_2)^{-1} & \cdots & \beta_n G(\beta_n)^{-1} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ \beta_1^{r-1}G(\beta_1)^{-1} & \beta_2^{r-1}G(\beta_2)^{-1} & \cdots & \beta_n^{r-1}G(\beta_n)^{-1} \end{bmatrix}$$

作为校验矩阵.

Goppa 码是线性码. 码字长 n，码间最短距离 $d \geqslant r+1$，r 是 $G(z)$ 的次方. 明文的长度 $l \geqslant n - mr$.

例 2　取$G(z) = z^2 + z + 1$

$$L = GF(2^3) = \{0, 1, \beta, \beta^2, \cdots, \beta^6\}$$

$p=2$，$p^m=2^3=8$. 取不可化约的多项式 $p(z)=z^3+z+1$，β 是$GF(2^3)$的生成元素，可以证明 L 中的元素均满足 $G(\beta_i) \neq 0$.

$\because$　$\beta^3=\beta+1$，　$\beta^6=\beta+\beta^2$

$\beta^5=\beta^2+\beta^3=1+\beta+\beta^2$

$\beta^2=\beta+\beta+\beta^3=1+\beta^2$，　$\beta^7=\beta+\beta^6=1$

$\beta^3=\beta$

β 表示	二进制表示	多项式表示
0	0 0 0	0
1	1 0 0	1
β	0 1 0	x
β^3	0 0 1	x^2
β^3	1 1 0	$1+x$
β^4	0 1 1	$x+x^2$
β^3	1 1 1	$1+x+x^2$
β^3	1 0 1	$1+x^2$

$G(x)=x^2+x+1$

$$G(0)=1$$

$$G(1)=1$$

$$G(\beta)=\beta^2+\beta+1=\beta^5$$

$$G(\beta^2)=\beta^5+\beta^2+1=1+\beta=\beta^2$$

$$G(\beta^3)=\beta^5+\beta^3+1=1+\beta^2+1+\beta+1=1+\beta+\beta^2=\beta^5$$

$$G(\beta^4)=\beta^5+\beta^4+1=\beta+\beta+\beta^2+1=1+\beta^2=\beta^2$$

$$G(\beta^5)=\beta^{10}+\beta^5+1=\beta^5+1+\beta+\beta^2+1$$
$$=1+\beta+1+\beta+\beta^2+1=\beta^2$$

$$G(\beta^2)=\beta^{12}+\beta^2+1=\beta^2+\beta^2+1$$
$$=1+\beta+\beta^2+1+\beta^2+1$$
$$=1+\beta=\beta^2$$

$\because \quad \beta^7=1$

$$\therefore \quad H=\begin{pmatrix} \frac{1}{1} & \frac{1}{1} & \frac{1}{\beta^2} & \frac{1}{\beta^3} & \frac{1}{\beta^5} & \frac{1}{\beta^2} & \frac{1}{\beta^2} & \frac{1}{\beta^3} \\ \frac{0}{1} & \frac{1}{1} & \frac{\beta}{\beta^5} & \frac{\beta^2}{\beta^3} & \frac{\beta^2}{\beta^2} & \frac{\beta^4}{\beta^2} & \frac{\beta^5}{\beta^3} & \frac{\beta^6}{\beta^2} \end{pmatrix}$$

$$=\begin{pmatrix} 1 & 1 & \beta^2 & \beta^4 & \beta^2 & \beta & \beta & \beta^4 \\ 0 & 1 & \beta^3 & \beta^3 & \beta^2 & \beta^2 & \beta^3 & \beta^3 \end{pmatrix}$$

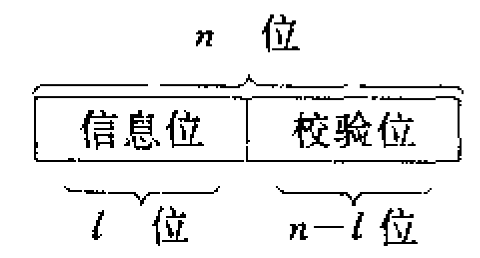
$$=\begin{pmatrix} 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1 \\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1 \\ \cdots\cdots \\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1 \\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0 \end{pmatrix}$$

§ 15 基于编码的公钥密码

默克尼斯（McEliece）于 1978 年提出一种利用郭帕码构造公钥密码的方案. 设郭帕码能纠正 t 个错误. 明文为 l 位的 0，1 符号串，设 $m=m_1m_2\cdots m_l$

生成矩阵 $G=(\boldsymbol{E}_{(l)} \vdots \boldsymbol{A})_{l\times n}$

其中 $\boldsymbol{E}_{(l)}$ 为 l 阶单位矩阵，所以郭帕码将 l 位明文变换为 n 位的码字，码字前 l 位是明文，后面 $n-l$ 位是校验位.



其中 $n=2^k$. 用户 A 构造一郭帕码的 $l_A\times n_A$ 阶生成矩阵 G_A. 用户 A 随机选一个 $l_A\times l_A$ 阶非奇异矩阵 S_A 及 $n_A\times n_A$ 的排列矩阵 P_A，记

$$G_A^*=S_AG_AP_A$$

用户 A 将 G_A^* 公开，而将 S_A，G_A 和 P_A 保密.

若用户 B 要向 A 送去信息 m，设 m 为 l_A 位的 0，1 符号串. 加密算法是：

$$c=mG_A^*+e$$

e 是权小于等于 t_A 由发方随机选取的 n 维向量.

解密算法是先作

$$cP_A^{-1}=mS_AG_AP_AP_A^{-1}+eP_A^{-1}=(mS_A)G_A+eP_A^{-1}$$

由于 eP_A^{-1} 的权 $\leqslant t_A$，故郭帕码能够恢复 mS_A，进而对 mS_A 右乘以 S_A^{-1}，便得到明文 m.

对默克尼斯公钥体制最有效的攻击方法是码译者从密文 c 任选 l 个分量，令由这些 l 比特分量构成的向量为 C_l，G_l^* 为 G^* 中相对应的 l 列组成的矩阵，e_l 为 e 中相应的 l 位组成的向量，则

$$C_l = mG_l^* + e_l$$

若 G_l^* 可逆，则 $(C_l + e_l)(G_l^*)^{-1} = m$. 若 $e_l = 0$，则

$$m = C_l(G_l^*)^{-1}$$

因此求得 m. 破译者随机选取 l 个分量使 $e_l = 0$ 的概率 P 为 $\binom{n-t}{l} \Big/ \binom{n}{l}$，而 G_l^* 求逆的工作量为 $O(l^3)$，因而求出 m 所需要的工作因子为 l^3/p，以默克尼斯建议的 $n = 1024 = 2^{10}$，$t = 50$，$l \approx n - mt = 1024 - 50 \times 10 = 524$，破译的工作因子大约为 $2^{80.7} \approx 1.96 \times 10^{24}$.

默克尼斯公钥体制的优点在于加解密算法简单快速，比 RSA 公钥体制快得多，安全性也很高，但公钥量太大. 有人曾经计算出 $t = 37$ 时，$l = 654$，这时安全性达到最高，工作因子为 $2^{84.1} \approx 2.07 \times 10^{25}$，公钥量达 6×10^5 比特.

§16 概率加密

以前讨论的公钥密码系统，当密钥确定之后，明文和密文对应关系是确定的，所以叫做确定型公钥密码系统. 公钥密码系统多少会给敌人提供若干信息，特别是在明文空间不十分大时.

(1)1982 年，勾德哇塞(Goldwasser)和密卡里(Micali)提出了概率加密的概念.

设 n 是某一正整数，$\left(\frac{x}{n}\right)$ 表示 $x \pmod n$ 的雅科比（Jacobi）符号，即若 $n=p_1^{\alpha_1}p_2^{\alpha_2}\cdots p_k^{\alpha_k}$，其中 p_i 是素数，则

$$\left(\frac{a}{n}\right)=\left(\frac{a}{p_1}\right)^{\alpha_1}\left(\frac{a}{p_2}\right)^{\alpha_2}\cdots\left(\frac{a}{p_k}\right)^{\alpha_k}$$

$\left(\frac{a}{p_i}\right)$ 叫做 $a \pmod{p_i}$ 的勒让德(Legendre) 符号，即

$$\left(\frac{a}{p_i}\right)=\begin{cases}1 & p_i\nmid a \quad 若 a 是模 p_i 的平方剩余\\ 0 & 若 p_i \mid a.\\ -1 & p_i\nmid a \quad 若 a 不是模 p 的平方剩余.\end{cases}$$

令

$$Z_n^* \triangleq \{x \mid 1\leqslant x<n,\ (x,\ n)\ =1\}$$

$$Z_n' \triangleq \left\{x \mid 1\leqslant x<n,\ \left(\frac{x}{n}\right)=1\right\}$$

$$Q_n(x) \triangleq \begin{cases}1, & 若 x 是 \bmod n 的平方剩余.\\ 0, & 若 x 不是 \bmod n 的平方剩余.\end{cases}$$

n 是素数，或已知所有 n 的因数时，求 $Q_n(x)$ 不难，否则是很困难的.

勾德哇塞和密卡里的概率加密公钥系统如下：

(a) 每一用户(设为 A) 给定安全参数 k(正偶数)，并随机选取两个不同的素数 p 和 q，$|p| = |q| = k/2$.

(b) 令 $n=pq$.

(c) 选 $y\in Z_n'$，y 不是模 n 的平方剩余，用户将 $\{n,\ y\}$ 公开，但 p，q 保密.

设用户 B 要向 A 发送信息 m，

$$m=m_1m_2\cdots m_l,\quad m\in \{0,\ 1\}^l$$

加密步骤：

S1：对 $i=1,\ 2,\ \cdots,\ l$，执行 S_2—S_3；

S2：随机取 $x\in Z_n^*$；

S3：若 $m_i=1$，则令 $e_i \equiv x^2 \pmod n$．否则，令

$$e_i \equiv x^2 y \pmod n$$

S4：B 将 $e_1e_2\cdots e_l$ 传送给 A.

显然 m 的密文不再是确定的，而依赖于 x 的选取. 解密步骤：A 收到密文 $e_1e_2\cdots e_l$ 后作：

S1′：对 $i=1, 2, \cdots, l$ 执行 $S2'$；

S2′：$m_i = Q_n(e_i)$；

S3′：得明文 $m=m_1m_2\cdots m_l$.

可以证明上述的概率加密有很高的安全性，但是数据的膨胀率为 k，即 1 个比特的明文对应于 k 个比特的密文，所以实际上它是不能接受的.

（2）L·布隆，M·布隆，M·索布（L. Blum，M. Blum，M. Shub）等人提出另一种概率加密算法，如下：设 p、q 是一对互异的素数，$|p|=|q|=k/2$，$p\equiv q\equiv 3 \pmod 4$ 这样的素数称为布隆数.

令 $n=pq, m=m_1m_2\cdots m_l \in \{0,1\}^l$ 为明文，加密过程是

S1：随机选 $x \in Z_n^*$；

S2：令 $x_0 \equiv x^2 \pmod n$；

S3：对 $i=1,2,\cdots,l$ 执行 S_4—S_5；

S4：$x_i \equiv x_{i-1}^2 \pmod n$；

S5：$b_i = x_i$ 的最后一位；

S6：$x_{l+1} \equiv x_l^2 \pmod n$；

S7：密文 $E(m)=(m \oplus b, x_{l+1})$，其中 $b \doteq b_1b_2\cdots b_l$，$\oplus$为模 2 加.

解密过程. 由 x_{l+1} 逐次恢复 $x_l, x_{l-1}, \cdots, x_1$，由之获得 $b=b_1b_2\cdots b_l$，与 $E(m)$ 的第 1 分量作$\oplus$运算可获得 m. 此时膨胀率为 $1+k/l$.

§ 17　素数的概率判定法

（1）关于素数的研究已有相当长的历史，只是由于近代密码学的研究才给它注入了新的动力，提出了新的课题. 其中重要的一个是素数的判定. 当然还有大数的因子分解. 近年来在这些方面都有新的进展. 在这里进行详细讨论是不可能的，它需要很多数论的基础.

下面将介绍实用的概率测试算法.

在进行测试之前必须将合数过滤掉，特别是大素数分布具有稀疏的特点，所以过滤显得十分重要. 关于素数分布有一个重要的定理.

定理 2.17.1　设 $\pi(x)$ 为小于或等于 x 的全部素数个数，则

$$\lim_{x\to\infty}\pi(x)\Big/\frac{x}{\ln x}=1$$

从这个定理可知 x 充分大时

$$\pi(x)\approx\frac{x}{\ln x}$$

利用这个公式可以估计在 $10^{99}\sim10^{100}-1$ 这一区间上素数的个数. 即 100 位 10 进制数中素数个数 N.

$$N\approx\frac{10^{100}}{\ln 10^{100}}-\frac{10^{99}}{\ln 10^{99}}=\frac{10^{100}}{100\ln 10}-\frac{10^{99}}{99\ln 10}$$

$$=\frac{10^{99}}{\ln 10}\left(\frac{10}{100}-\frac{1}{99}\right)\approx\frac{9\cdot 10^{99}}{100\ln 10}$$

100 位 10 进制数的个数为

$$10^{100}-10^{99}-1\approx10^{99}\ [10-1]\ \approx9\cdot10^{99}$$

故 100 位 10 进制数的素数密度为 $\frac{1}{100\ln 10}$.

一般可证 n 位 10 进制数中素数的密度当 n 充分大时约为 $\frac{1}{n\ln 10}$. 素数的稀疏性从此可知. 一般说来判定一个数是素数比较难，而否定它是素数要容易得多. 比如偶素数只有 2 这一个，故偶数应排除. 同样最后一位为 5 的数也不是素数. 我们希望有一个简便而有效的算法能将大量合数"淘汰"出去，对余下的数再进行素数判定，好在素数的研究在数论中是比较成熟的课题.

对素数判断有一个充要条件，叫做威尔逊（Wilson）定理.

定理 2.17.2 n 是素数的充要条件是

$$(n-1)! \equiv -1 (\bmod n)$$

证明：(1) 必要性.

n 是素数，所以在 $\{1, 2, \cdots, n-1\}$ 中每一个元素 a 都有一元素 a^{-1}，使得 $aa^{-1} \equiv 1 (\bmod n)$，除 1 和 $n-1$ 这两个数它的逆元素为自身外，即

$$1 \cdot 1 \equiv 1 (\bmod n), \quad (n-1) \cdot (n-1) \equiv (\bmod n)$$

其它 $n-3$ 个数，$a^{-1} \neq a$，故成对出现 $aa^{-1} \equiv 1$ $(\bmod n)$.

$$(n-1)! \equiv (n-1) \equiv -1 (\bmod n).$$

(2) 充分性. 用反证法.

设 n 不是素数. 设 $n = a \cdot b$，其中 $a, b > 1$，则 $a \mid (n-1)!$ 同样理由 $a \mid [(n-1)! + 1]$，故 $a \mid [(n-1)! + 1 - (n-1)!]$，但 $[(n-1)! + 1] - (n-1)! = 1$，这跟 $a > 1$ 的假设矛盾. 这就证明了 n 是素数. 证毕.

(2) 威尔逊定理给出判定素数的充要条件，有很高的理论价值.

关于素数还有一个费尔玛(Fermat)定理. 此定理给出素数 p 的必要条件，若不满足，则可断定它不是素数.

定理是这样的：若 p 是素数，则对于任意的整数 a，应有 $a^{p-1}\equiv 1\pmod p$.

下面介绍的勒宾（Rabin）素数概率测试法实际上是基于费尔玛定理．当然满足费尔玛定理的数 p 不能判定它是素数，但不满足这定理的数 p 肯定不是素数，因此，它可以起到过滤作用．

令 $n-1=2^l m$，其中 l 是非负整数，m 是正奇数．若 $b^m\equiv 1\pmod n$ 或 $b^{2^j m}\equiv -1\pmod n$，$0\leqslant j\leqslant l-1$，则称 n 通过以 b 为基的密勒（Miller）测试．

定理 2.17.3 若 n 是素数，b 是正整数，且 $n\nmid b$，则 n 必然通过以 b 为基的密勒测试．

证：令

$$S_k\equiv b^{(n-1)/2^k}\pmod n\equiv b^{2^{(l-k)}m}\pmod n$$

$$k=l,\ l_{-1},\ \cdots,\ 2,\ 1,\ 0$$

其中 $S_l\equiv b^m\pmod n$，$S_0\equiv b^{n-1}\pmod n$

若 n 是素数，则由费尔玛定理说明 $S_0\equiv b^{n-1}\equiv 1\pmod n$ 必然成立．它的必然结果是

$$S_1\equiv b^{(n-1)/2}\equiv 1\pmod n \text{ 或 } S_1\equiv -1\pmod n$$

必然成立．而且 $S_1\equiv 1\pmod n$ 或 $S_1\equiv -1\pmod n$ 成立时，费尔玛定理必然成立．因 $S_0\equiv S_1^2\equiv 1\pmod n$．同理，若 $S_1\equiv 1\pmod n$ 或 $S_2\equiv -1\pmod n$，则 $S_1\equiv 1\pmod n$，因而满足费尔玛定理．

依此类推，若已知

$$S_{k+1}\equiv 1\pmod n \quad \text{或} \quad S_{k+1}\equiv -1\pmod n$$

则 $\quad S_k\equiv S_{k-1}\equiv\cdots\equiv S_0\equiv 1\pmod n$，

即费尔玛定理成立．

定理说明密勒测试一旦通过，费尔玛定理便可满足．在计

算 S_k 的过程中，从 $k=l, l-1, l-2, \cdots, 2, 1, 0$ 依此进行，最终为 S_0 即为 $b^{s-1} \pmod{n}$.

下一个定理证明比较繁琐，可见于有关的数论书，这里仅仅叙述其结果.

定理 2.17.4 若 n 是奇合数，则 n 通过以 b 为基的密勒测试的 b 的数目最多为 $(n-1)/4, 1\leqslant b\leqslant n-1$.

基于上述两个定理，若 n 是正整数，选 k 个小于 n 的正整数，以这 k 个数作为基进行密勒测试，若 n 是合数，k 次测试全部通过的概率为 $\left(\frac{1}{4}\right)^k$.

比如 $k=100$，n 是合数，但测试全部通过的概率为 $\frac{1}{4^{100}}=6.22\times10^{-61}$，这是很小的数，说明这样的事情几乎不可能发生.

§18 科尔—列维斯特(Chor-Rivest)背包公钥密码系统

(1) 班利·科尔 (Benny Chor) 和罗纳德·列维斯特 (Ronald L. Rivest) 于 1985 年发表一篇名叫"A Knapsack Type Public Key Cryptosystem Based on Arithmetic in Finite Fields"的文章中提出一种新型的背包公钥系统，以前的低密度攻击法对于它失去了作用.

背包问题的解不仅困难，还可能不唯一. 但超递增序列的背包问题没有这问题. 科尔—列维斯特背包公钥系统与之相应的依据是布斯—邹拉定理，定理如下：

定理（布斯—邹拉：Bose-Chowla） 设 p 是素数，n 是不低于 2 的整数，则存在序列

$$A=\{a_i \mid 0\leqslant i\leqslant p-1,\ a_i\in \mathbf{Z}\}$$

使得 (a) $1 \leqslant a_i \leqslant p^n - 1$，$i = 0, 1, 2, \cdots, p-1$

(b) $\xi = (x_0, x_1, \cdots, x_{p-1})$，$\eta = (y_0, y_1, \cdots, y_{p-1})$ 是两个坐标为非负整数的不同向量，且若

$$\sum_{i=0}^{p-1} x_i = \sum_{j=0}^{p-1} y_j \leqslant n$$

则

$$\sum_{i=0}^{p-1} a_i x_i \neq \sum_{j=0}^{p-1} a_j y_j$$

成立.

证：证明是构造出序列 A.

通过选择素数 p 及在 $GF(p)$ 域上不可化约多项式 $I(x)$，构造域 $GF(p^n)$. 设 $t \in GF(p^n)$，且是 $GF(p)$ 域上 n 次最小多项式的根. 又设 $g \in GF(p^n)$ 是 $GF(p^n)$ 上乘法群的母元素.

$$t + GF(p) = \{t + i \in GF(p^n) \mid i = 0, 1, 2, \cdots, p-1\}$$

显然 $\quad t + GF(p) \subseteq GF(p^n)$

设

$$a_i = \log_g(t + i), \quad i = 0, 1, 2, \cdots, p-1$$

则 $\{a_i\}_{i=0}^{p-1}$ 便是所求的序列. 因为 g 是 $GF(p^n)$ 的乘法群的母元素，所以 g 有 $p^n - 1$ 个不同指数，故 a_i 的全体都在 $[1, p^n - 1]$ 区间上.

由于 $(x_0, x_1, \cdots, x_{p-1})$ 和 $(y_0, y_1, \cdots, y_{p-1})$ 是两个坐标为非负整数的不同向量，而且 $\sum_i x_i, \sum_j y_i \leqslant n$.

若 $\quad \sum_{i=0}^{p-1} a_i x_i = \sum_{j=0}^{p-1} a_j y_j$

则 $\quad g^{\sum_i a_i x_i} = g^{\sum_j a_j y_j}$

$\therefore \quad \prod_i g^{a_i x_i} = \prod_j g^{a_j y_j}$

但 $\quad g^{a_i}=t+i$

$\therefore\quad (t+i_1)^{x_1}(t+i_2)^{x_2}\cdots(t+i_h)^{x_h}$

$\quad =(t+j_1)^{y_1}(t+j_2)^{y_2}\cdots(t+j_k)^{y_k}$

其中$(i_1,i_2,\cdots,i_h)$和$(j_1,j_2,\cdots,j_k)$是来自$\{0,1,\cdots,p-1\}$的非空集合. 由于

$$\sum_i x_i=\sum_j y_j\leqslant n$$

故每个集合至多 n 个元素.

由于向量$\xi\neq\eta$，所以

$$\prod_{p=1}^{h}(t+i_p)^{x_p}-\prod_{q=1}^{k}(t+j_q)^{y_q}$$

是系数在$GF(p)$上的非零多项式，而且次方$\leqslant n-1$. 首次系数为 1 相消，故至少降 1 次方. 这跟假定 t 是$GF(p)$上是 n 次最小多项式的根的假定相矛盾.

(2) 建立科一列密码系统的第一步是挑选 p 和 n，进而在$GF(p^n)$上选取次方为 n 的 $t\in GF(p^n)$和 $g\in GF(p^n)$. t 和 g 的选择有较大的自由度. 根据布斯一邹拉定理计算$GF(p)+t$ 中 p 个元素以 g 为底的对数，现将密码系统产生的过程叙述如下：

(a) 选择素数 p 及整数 $n\leqslant p$，使得在$GF(p^n)$中的离散对数能有效地计算.

(b) 通过选择在$GF(p)$上不可化约的 n 次多项式$I(x)$构造$GF(p^n)$，也就是由 mod $I(x)$ 来表示$GF(p^n)$.

(c) 随机选择$t\in GF(p^n)$使得t是$GF(p)$上 n 次极小多项式的根.

(d) 随机选择 $g\in GF(p^n)$，要求它是$GF(p^n)$的乘法群的母元素.

(e) 计算 $a_i=\log_g(t+i)$，$\quad i=0,1,2,\cdots,p-1$

(f) 随机选择一 p 个元素的置换 π，使得

π: $a_i \to b_i = a_{\pi(i)}$, $i=0, 1, \cdots, p-1$

(g) 选一随机数 d，$0 \leqslant d \leqslant p^n-2$，令

$c_i = b_i + d$，$i=0, 1, 2, \cdots, p-1$

(h) 将 $c_0, c_1, c_2, \cdots, c_{p-1}$ 及 p、n 公开.

(i) t，g，π，d 保密.

加密算法：

科尔一列维斯特系统适合于对权正好为 n 的 p 位 0,1 字符串明文块进行加密. 因此首先要建立表达 $[0, c(p, n)]$ 中整数的位串到长度 p，权 n 的块变换.

对权 n 的位串 m 加密如下：

$$E(m) = c_{i_1} + c_{i_2} + \cdots + c_{i_n} (\bmod (p^n - 1))$$

其中 $i_1, i_2, \cdots, i_n$ 是 n 个 1 所在的位.

解密算法：

(a) 令 $r(t) \equiv t^n (\bmod I(t))$，$r(t)$ 是方次 $\leqslant n-1$ 的多项式.

(b) 设 $c = E(m)$，计算 $c' \equiv c - nd (\bmod (p^n - 1))$

(c) 计算 $p(t) \equiv g^{c'} (\bmod I(t))$，$p(t)$ 为 t 的 $n-1$ 次方的多项式.

(d) 计算 $d(t) = t^n + p(t) - r(t)$，$d(t)$ 是系数在 $GF(p)$ 上 t 的 n 次多项式.

(e) 将 $d(t)$ 因子分解得

$$d(t) = (t + i_1)(t + i_2) \cdots (t + i_n)$$

这是因为

$$\begin{aligned} g^{E(m)-nd} &= g^{c_{i_1}} g^{c_{i_2}} \cdots g^{c_{i_n}} g^{-nd} \\ &= g^{(c_{i_1}-d)} g^{(c_{i_2}-d)} \cdots g^{(c_{i_n}-d)} \\ &= g^{b_{i_1}} g^{b_{i_2}} \cdots g^{b_{i_n}} \end{aligned}$$

其中 b_{i_j} 是 $t + GF(p)$ 中某些元素的离散对数，所以考虑 $d(t)$ 是这样一些线性因子的积.

(f) 通过置换找到 n 个根 i_j，由 π^{-1}找到原来为 1 的坐标.

(3) **举例** 取 $p=5, n=3$. 不可化约的多项式 $I(x)=x^3+x+1$，在 $GF(5^3)$域上有

$$x^3=-x-1$$

$$x^4=-x^2-x$$

$$x^5=-x^3-x^2=-x^2+x+1$$

$$x^{10}=(-x^2+x+1)^2=x^4-2x^3-x^2+2x+1$$
$$=-x^2-x+2x+2-x^2+2x+1=-2x^2+3x+3$$

$$x^{15}=(-2x^2+3x+3)(-x^2+x+1)=2x^4-x^2+x+3$$
$$=-2x^2-2x-2x^2+x+3=x^2-x+3$$

$$x^{20}=(-2x^2+3x+3)^2=4x^4-2x^3-3x^2+3x+4$$
$$=-4x^2-4x+2x+2-3x^2+3x+4$$
$$=-2x^2-4x+1$$

$$x^{30}=(-2x^2+3x+3)(-2x^2-4x+1)$$
$$=4x^4+2x^3-20x^2-9x+3$$
$$=-4x^2-4x-2x-2-4x+3$$
$$=x^2+1$$

$$x^{34}=(x^2+1)(-x^2-x)=-x^4-x^3-x^2-x$$
$$=x^2+x+x+1-x^2-x=x+1$$

∵ $x^{31}=x^3+x=-x-1+x=-1$

∴ $x^{62}=1$

令 $g=2x^2+x+2$

$$g^2=(2x^2+x+2)^2=+4x^4+4x^3+4x^2+4x+4$$
$$=-4x^2-4x-4x-4+4x^2+4x+4$$
$$=-4x=x$$

故有 $x^{30}=g^{60}=x^2+1$

$$g^{62}=-1,$$

$g^{124}=1.$

$g^{68}=g^{62}\cdot g^{6}=-x^{3}=x+1$

$g^{98}=g^{68}\cdot g^{30}=(x+1)(x^{2}-x+3)$

$\quad=x^{3}+2x+3=-x-1+2x+3=x+2$

$g^{86}=g^{62}\cdot g^{24}=-(x^{12})=-(-2x^{2}+3x+3)x^{2}$

$\quad=2x^{4}-3x^{3}-3x^{2}=-2x^{2}-2x+3x+3-3x^{2}$

$\quad=x+3$

$g^{109}=g^{98}g^{11}=(x+2)x^{5}(2x^{2}+x+2)$

$\quad=(x+2)(-x^{2}+x+1)(2x^{2}+x+2)$

$\quad=(-x^{3}-x^{2}+3x+2)(2x^{2}+x+2)$

$\quad=(-x^{2}+4x+3)(2x^{2}+x+2)$

$\quad=-2x^{4}+7x^{3}+3x^{2}+x+1$

$\quad=2x^{2}+2x-2x-2+3x^{2}+x+1=x-1=x+4$

即 $\quad g^{2}=x,\quad g^{68}=x+1,\quad g^{98}=x+2$

$\quad g^{86}=x+3,\quad g^{109}=x+4$

令 $\quad a_0=2,\quad a_1=68,\quad a_2=98,\quad a_3=86,\quad a_4=109$

令 $\quad b_0=98,\quad b_1=109,\quad b_2=2,\quad b_3=68,\quad b_4=86$

$\quad c_0=109,\quad c_1=120,\quad c_2=13,\quad c_3=79,\quad c_4=97$

对于 $m=1\ 0\ 1\ 0\ 1$，$E(m)=109+13+97=219=c$

$c'=c-33=186$

$g^{c'}=g^{186}=g^{124}\cdot g^{62}=g^{62}=g^{60}\cdot g^{2}$

$\quad=(x^{2}+1)x=x^{3}+x=-x-1+x=-1.$

$x^{3}+x+1+-1=x^{3}+x=x(x^{2}+1)=x(x-2)(x-3)$

$\quad=x(x+3)(x+2)$

故

$$x(x+2)(x+3)=g^{2}g^{98}g^{86}=g^{b_2}g^{b_0}g^{b_4}$$

故解密得明文 $m=1\ 0\ 1\ 0\ 1.$

又如　$b_0=86$，　$b_1=109$，　$b_2=68$，　$b_3=2$，　$b_4=98$

　　　$c_0=98$，　$c_1=121$，　$c_2=80$，　$c_3=14$，　$c_5=110$

已知　$m=1\ 1\ 0\ 0\ 1$

$$c=E(m)=98+121+110=329$$

$$c'=329-36=293$$

$$\begin{aligned}g^{c'}&=g^{45}=g^{44}g=x^{22}g=(-2x^2-4x+1)x^2g\\&=(-2x^4-4x^3+x^2)(2x^2+x+2)\\&=(2x^2+2x+4x+4+x^2)(2x^2+x+2)\\&=(3x^2+x+4)(2x^2+x+2)\\&=x^4+x+3=-x^2-x+x+3=x^2+3\end{aligned}$$

$$\begin{aligned}x^3+x^2+x+4&=(x-1)(x-2)(x-3)\\&=(x+2)(x+3)(x+4)=g^{a_2}g^{a_3}g^{a_4}\\&=g^{b_4}g^{b_0}g^{b_1}\end{aligned}$$

故　$m=1\ 1\ 0\ 0\ 1$

§19　离散对数问题

在第 2 章 §2 中给出了已知 x 及 r 求 $x^r \pmod n$ 的算法，它的反问题是已知 $y=x^r \pmod n$ 及 x 求指数 r，我们称这反问题为离散对数．求离散对数并非易事，密码学中有许多办法就是利用它的困难性．例如狄非—赫尔曼公钥系统便是一例．用户 A 可以随机地选取 n_A，$1\leqslant n_A\leqslant p-2$，计算：

$$k_A\equiv a^{n_A} \pmod p$$

将 k_A 公开但 n_A 保密，这便是基于求离散对数的困难．用户 B 欲与 A 通信，他们的通信密钥

$$k_{AB}\equiv (k_A)^{n_B} \pmod p\equiv a^{n_An_B} \pmod p$$

$\equiv (k_B)^{x_A} \pmod p$

（1）在介绍求离散对数的算法之前，先讨论有名的中国剩余定理. 即：

已知 $m_1, m_2, \cdots, m_k$ 是两两互素的正整数，则同余方程且 $x \equiv a_i \pmod{m_i}$，$i=1, 2, \cdots, k$，模 $m_1 m_2 \cdots m_k$ 有唯一解.

下面给出求解的步骤. 令

$$M = \prod_{i=1}^{k} m_i = m_1 m_2 \cdots m_k$$

$$M_j = M/m_j = \prod_{\substack{i=1 \\ i \neq j}}^{k} m_j$$

令 y_j 满足

$$M_j y_j \equiv 1 \pmod{m_j}, \quad j=1, 2, \cdots, k$$

由于$(M_j, m_j)=1$，故 y_j 存在，不难证明

$$x \equiv a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_k M_k y_k \pmod M$$

便是问题的解，而且是唯一的.

例

$$x \equiv 1 \pmod 2, \qquad x \equiv 2 \pmod 3$$

$$x \equiv 3 \pmod 5, \qquad x \equiv 5 \pmod 7$$

$$M = 2 \cdot 3 \cdot 5 \cdot 7 = 210,$$

$$M_1 = 105, \quad M_2 = 70, \quad M_3 = 42, \quad M_4 = 30.$$

$$M_1 y_1 \equiv 1 \pmod 2, \quad y_1 = 1$$

$$M_2 y_2 \equiv 1 \pmod 3, \quad y_2 = 1$$

$$M_3 y_3 \equiv 1 \pmod 5, \quad y_3 = 3$$

$$M_4 y_4 \equiv 1 \pmod 7, \quad y_4 = 4$$

$$\therefore \quad x \equiv 105 + 2 \times 70 + 3 \times 42 \times 3 + 5 \times 30 \times 4 \pmod{210}$$

$$\equiv 173$$

（2）下面介绍波涅格—赫尔曼（Pohlig-Hellman）求离散对数的方法．已知 x 和 y 都是正整数，且 x 是 $GF(p)$ 域上的本原元素．求正整数 r，满足

$$y \equiv x^r \pmod{n} \qquad (2.19.1)$$

假定整数 n 为素数 p，且 $p-1$ 只有小素数因子．

设

$$p-1 = p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k} \quad n_i > 0,\ 1 \leqslant i \leqslant k$$

根据中国剩余定理，只要能确定

$$r_i \equiv r \pmod{p_i^{n_i}} \qquad 1 \leqslant i \leqslant k$$

则 r 便能求得．下面讨论求 r_i 的方法．以求 r_1 为例，求 r_2，r_3，…，r_k 类似可求得．

若

$$M_i = (p-1)/p_i^{n_i}, \qquad i = 1,2,\cdots,k$$

$$M_i y_i \equiv 1 \pmod{p_i^{n_i}}, \qquad i = 1,2,\cdots,k$$

则

$$r = r_1 M_1 y_1 + r_2 M_2 y_2 + \cdots + r_k M_k y_k$$

故

$$y^{(p-1)/p_1} \equiv (x^r)^{(p-1)/p_1} \pmod{p}$$

$$\equiv (x^{r_1 M_1 y_1 + r_2 M_2 y_2 + \cdots + r_k M_k y_k})^{(p-1)/p_1} \pmod{p}$$

$$\equiv (x^{r_1 M_1 y_1})^{(p-1)/p_1} (x^{r_2 M_2 y_2})^{(p-1)/p_1} \cdots (x^{r_k M_k y_k})^{(p-1)/p_1} \pmod{p}$$

由于 $M_1 y_1 \equiv 1 \pmod{p_1^{n_1}}$，令 $M_1 y_1 = h p_1^{n_1} + 1$

$$\therefore \quad (x^{r_1 M_1 y_1})^{(p-1)/p_1} \equiv (x^{r_1 (h p_1^{n_1} + 1)})^{(p-1)/p_1} \pmod{p}$$

$$\equiv (x^{r_1 h p_1^{n_1 - 1}})^{p-1} (x^{r_1})^{(p-1)/p_1} \pmod{p}$$

由于 $\qquad x^{p-1} \equiv 1 \pmod{p}$

$$\therefore \quad (x^{r_1 M_1 y_1})^{(p-1)/p_1} \equiv (x^{r_1})^{(p-1)/p_1} \pmod{p}$$

另一方面 $p_1 | M_j, j = 2,3,\cdots,k$，故 $j \neq 1$ 时，

$$(x^{r_j M_j y_j})^{(p-1)/p_1} \equiv (x^{(p-1)})^{r_j M_j y_j / p_1} \equiv 1 \quad \pmod{p}$$

所以

$$y^{(p-1)/p_1} \equiv (x^r)^{(p-1)/p_1} \equiv (x^{r_1})^{(p-1)/p_1} (\text{mod } p)$$

$\because$ $$y^{(p-1)/p_1} \equiv (x^{r_1})^{(p-1)/p} (\text{mod } p) \quad (2.19.2)$$

$$r_1 = r_{10} + r_{11}p_1 + r_{12}p_1^2 + \cdots + r_{1(n_1-1)}p_1^{n_1-1}$$

$$0 \leqslant r_{1j} \leqslant p_1 - 1, \quad j = 1, 2, \cdots, n_1 - 1.$$

$\therefore$ $$y^{(p-1)/p_1} \equiv x^{r_{10}(p-1)/p_1} (\text{mod } p) \quad (2.19.3)$$

比较等号两端可以确定 r_{10}，但

$$x^{r_1 - r_{10}} = x^{r_{11}p + r_{12}p^2 + \cdots + r_{1(n_1-1)}p_1^{n_1-1}}$$

若令

$$y_1 = yx^{-r_{10}}$$

则有

$$(yx^{-r_{10}})^{(p-1)/p_1} = (x^{r_{11}(p-1)})(x^{r_{12}(p-1)p_1})\cdots(x^{r_{1(n_1-1)}(p-1)p_1^{n_1-2}})$$

$$\equiv (x^{(p-1)/p_1})^{r_{11}} (\text{mod } p) \quad (2.19.4)$$

由(2.19.4)等号两端可以确定 r_{11}. 类似理由令

$$y_2 = y_1 x^{-r_{11}p_1}$$

$\therefore$ $$(y_2)^{(p-1)/p_1^2} \equiv x^{r_{12}(p-1)/p_1}$$

由上式可确定 r_{12}. 依此类推可求得 $r_{13}, \cdots, r_{1(n_1-1)}$. 为了计算方便起见，令 $\beta_1 = x^{(p-1)/p_1}$，预先计算 $\beta_1, \beta_1^2, \cdots, \beta_1^{p_1-1}$.

(3) 举例. $p=8101$, $x=6$, $y=7833$. 求 r 使之满足

$$y \equiv x^r \ (\text{mod } p)$$

$$8101 = 6 \times 1350 + 1$$

$$6(-1350) \equiv 1 \quad (\text{mod } 8101)$$

$\therefore$ $$6^{-1} \equiv 6751 \quad (\text{mod } 8101)$$

对于 $p_1 = 2$ 有

$$\beta_1 \equiv 6^{8100/2} \equiv 6^{4050} \equiv 8100 \ (\text{mod } 8101)$$

β_1	β_1^2
8100	1

对于 $p_2=3$，有

$$\beta_2=6^{8100/3}=6^{2700}\equiv 5883\ (\text{mod } 8101)$$

β_2	β_2^2	β_2^3
5883	2217	1

对于 $p_3=5$，有

$$\beta_3\equiv 6^{8100/5}\equiv 6^{1620}\equiv 3547\ (\text{mod } 8101)$$

β_3	β_3^2	β_3^3	β_3^2	β_3^2
3547	356	7077	5221	1

$M_1y_1\equiv 1\ (\text{mod } 4)$，　　$M_1=3^45^2=1215$

$M_1y_1\equiv 0\ (\text{mod } 3^4)$，　　$M_1y_1\equiv 2025(\text{mod } 8101)$

$M_1y_1\equiv 0\ (\text{mod } 5^2)$，

$M_2y_2\equiv 0(\text{mod } 2^2)$，　　$M_2=2^25^2=100$

$M_2y_2\equiv 1(\text{mod } 3^4)$，　　$M_2y_2\equiv 6400(\text{mod } 8101)$

$M_2y_2\equiv 0(\text{mod } 5^2)$，

$M_3y_3\equiv 0(\text{mod } 2^2)$，　　$M_3=2^23^4=324$

$M_3y_3\equiv 0(\text{mod } 3^4)$，　　$M_3y_3\equiv 7776(\text{mod } 8101)$

$M_3y_3\equiv 1(\text{mod } 5^2)$，

(a) $p_1=2$，　$n_1=2$

$y=7833$，　$y^{8100/2}\equiv 8100$

$\therefore$　$r_{10}=1$，　$y_1=yx^{-1}\equiv 5356$

$(y_1)^{8100/2^2}\equiv 1(\text{mod } 8101)$

即　$r_{11}=0$

$\therefore \quad r_1 = 1$

(b) $p_2 = 3, \quad n_2 = 4$

$y = 7833, \quad y^{8100/3} \equiv 8100, \quad r_{20} = 2$

$y_1 = yx^{-2}, \quad 6^{-2} \equiv 7876 \pmod{8101}$

$\therefore \quad y_1 \equiv 7833 \times 7876 \equiv 3593 \pmod{8101}$

$(y_1)^{8100/3^2} \equiv (3592)^{900} \equiv 2217 \pmod{8101}$

$\therefore \quad r_{21} = 2$

$y_2 = y_1 x^{-6} \equiv 3593 \times 7482 \pmod{8101}$

$\equiv 3708$

$(y_2)^{8100/3^3} \equiv (3708)^{300} \equiv 5883 \pmod{8101}$

$\therefore \quad r_{22} = 1$

$y_3 = y_2 x$ [illegible] $\equiv 3708 \times 2960 \equiv 6926 \pmod{8101}$

$(y_3)^{8100/3^4} \equiv (6926)^{100} \equiv 5883 \pmod{8101}$

$\therefore \quad r_{23} = 1$

$\therefore \quad r_2 = 2 + 2 \times 3 + 3^2 + 3^3 = 44$

(c) $p_3 = 5, \quad n_3 = 2$

$y = 7833, \quad y^{8100/5} \equiv (7833)^{1620} \equiv 356 \pmod{8101}$

$\therefore \quad r_{30} = 2$

$y_1 = yx^{-2} \equiv 7833 \times 7876 \equiv 3593$

$(y_1)^{8100/5^2} \equiv 356 \pmod{8101}$

$\therefore \quad r_{31} = 2$

$r_3 = 2 + 2 \times 5 = 12$

由中国剩余定理可得

$r = r_1 M_1 y_1 + r_2 M_2 y_2 + r_3 M_3 y$，　即

$r = 2025 + 44 \times 6400 + 12 \times 7776 = 376937$

$\equiv 4337 \pmod{8101}$

当然波涅格—赫尔曼方法建立在 $p-1$ 可以因子分解成 $p_1^{r_1}p_2^{r_2}\cdots p_k^{r_k}$ 的基础上．也就是在 p_1，p_2，…，p_k 都是小素数时才可能．

§20 关于公钥密码的几点补充及密钥分存问题

公钥密码的思想无疑是先进的，但时间不长，还没有一个足够可靠的公钥系统可供使用．RSA 虽然很有希望，但处理 200 位 10 进制数的幂运算效率较低，公钥密码由于要将相当大一部分关于密码的信息予以公开，它势必对系统产生影响，为此要付出代价来补偿．

公钥一出现便显示出它的优势，有一种咄咄逼人取而代之的架势，实际上公钥还在发展中，它和传统密码学构成近代密码学的不可分割的组成部分．下面介绍将两者结合起来取长补短的一条蹊径．

(1) 假定存在一通信网络系统，这个系统有一密钥分配中心 KDC (Key Distribution Center)，它专门负责管理通信密钥，每一用户都在 KDC 存有各自的密钥．这些用户密钥 k_A，k_B，…，都用只有 KDC 掌握的密钥予以加密后保存．若用户 A 欲和用户 B 秘密通信，则 A 向 KDC 提出申请，为此送去信息 (A, B)．KDC 便随机地产生一通信密钥 k，用 A 和 B 的密钥 k_A 和 k_B 加密得 $k_A(k)$，$k_B(k)$ 并送给 A，A 收到后将 $k_B(k)$ 送给 B，密钥分配完毕，通信便可开始，这个过程如图 2.20.1 所示．

在这个系统中 KDC 是核心，它必须保证绝对安全，通过它利用传统密码建立起公钥密码．一切通信都要通过 KDC．但当系统较大时不胜其繁，也是一弊．

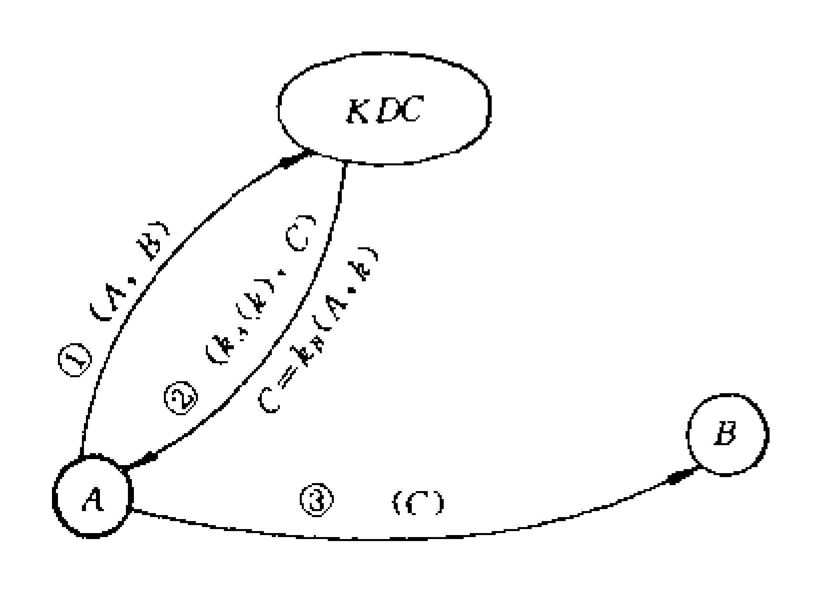


图 2-20-1

（2）混合密码体制.

公钥的长处在于密钥管理方便，但效率低，公钥之短正好是传统密码的长处，反之亦然. 如何取长补短建立起有效的公钥系统便是混合密码研究的核心思想，即通过公钥传送通信密钥，然后通过传统密码加密，以之达到保密通信的目的.

公钥密码还有一个不足之处，即若公钥文件被篡改，则公钥将被攻破，沙米尔建议以用户的身份(如姓名、性别等)作为公钥来克服这个缺点. 这样公钥便一成不变，这种方案也有其缺陷. 下面介绍一种秘密身份的验证方案以保证公钥的完整性，系统由两个公钥系统组成，它们的加密算法分别是 E_1 和 E_2，解密算法分别为 D_1 和 D_2. E_1 和 E_2 公开，但 D_1 和 D_2 保密，而且具有

$$E_i(X \cdot Y) = E_i(X) \cdot E_i(Y)$$

$$D_i(X \cdot Y) = D_i(X) \cdot D_i(Y), \qquad i = 1,2$$

显然 RSA 具有这种特性. "·"是乘法运算.

用户 A 进入系统时向 KDC 提供自己的身份 I_A 及公开的

加密密钥 k_A，KDC 计算

$$S_A = D_1(I_A)$$

$$T_A = D_1(E_2(k_A))$$

$$H_A = S_A \cdot T_A$$

H_A 称为 A 的秘密身份．KDC 将（I_A，k_A，H_A）列入公钥文件予以公布．到此 KDC 的任务便告结束，对以后的通信不进行干预．这是本系统的特点．A 欲和 B 通信时，可查得 B 的公钥（I_B，k_B，H_B）．B 计算

$$E_1(H_B)/E_2(k_B) = I_B^*$$

若 $I_B^* = I_B$，则 B 的公钥是完整的．

攻击者若改 k_B 为 k_B^*，它必须计算 H_B^*，使

$$E_1(H_B^*)/E_2(k_B^*) = I_B$$

$$\therefore \quad H_B^* = D_1(I_B \cdot E_2(k_B^*))$$

这只有掌握 D_1 才能做到．而只有 KDC 有此可能．反之，攻击者若先给出 H_B^*，以此来确定 k_B^*，使

$$E_1(H_B^*)/E_2(k_B^*) = I_B$$

即
$$k_B^* = D_2(E_1(H_B^*)/I_B)$$

这也是不可能做到的．因为只有 KDC 掌握 D_1 和 D_2 两种算法．

(3) 密钥分存问题．

若有 6 位科学工作者从事一项绝密研究，为了安全起见必需 4 位合作者到场方可开门，这样可以避免由某一位丢失钥匙而造成严重事故，若考虑门上加锁，则需要 $\binom{6}{3}=20$ 把锁，即 6 人中任意 3 人到场，至少有一把锁无法打开．每一合作者必须携带 $\binom{5}{3}=10$ 把钥匙，这种办法显然不现实，特别是人数更多

时，但可以考虑用 20 位 10 进制作为密钥，这样的密钥分别由 6 位合作者分开保管，每位保管其中 10 位，作为子密钥，使得任意 4 个子密钥可将密钥恢复，少于 4 人则不能，利用提到的思想设计如下：设密钥为 $a_1a_2a_3\cdots a_{19}a_{20}$，6 位合作者为 A_1，A_2，$\cdots$，A_6. 6 个人取 3 个组合有 $A_1A_2A_3$，$A_1A_2A_4$，$A_1A_2A_5$，$A_1A_2A_6$，$A_1A_3A_4$，$A_1A_3A_5$，$A_1A_3A_6$，$A_1A_4A_5$，$A_1A_4A_6$，$A_1A_5A_6$，$A_2A_3A_4$，$A_2A_3A_5$，$A_2A_3A_6$，$A_2A_4A_5$，$A_2A_4A_6$，$A_2A_5A_6$，$A_3A_4A_5$，$A_3A_4A_6$，$A_3A_5A_6$，$A_4A_5A_6$.

让这 20 种组合分别和 a_1，a_2，$\cdots$，a_{20}依次对应，比如 a_1 和 $A_1A_2A_3$ 对应，表示 A_1，A_2，A_3 都不掌握 a_1 这一位的信息，现将对应列表于下：

若令

$$A=\{a_1,a_2,a_3,a_4,\cdots,a_{19},a_{20}\}$$

$a_1\leftrightarrow A_1A_2A_3$，　$a_2\leftrightarrow A_1A_2A_4$，　$a_3\leftrightarrow A_1A_2A_5$，　$a_4\leftrightarrow A_1A_2A_6$，

$a_5\leftrightarrow A_1A_3A_4$，　$a_6\leftrightarrow A_1A_3A_5$，　$a_7\leftrightarrow A_1A_3A_6$，　$a_8\leftrightarrow A_1A_4A_5$，

$a_9\leftrightarrow A_1A_4A_6$，　$a_{10}\leftrightarrow A_1A_5A_6$，　$a_{11}\leftrightarrow A_2A_3A_4$，　$a_{12}\leftrightarrow A_2A_3A_5$，

$a_{13}\leftrightarrow A_2A_3A_6$，　$a_{14}\leftrightarrow A_2A_4A_5$，　$a_{15}\leftrightarrow A_2A_4A_6$，　$a_{16}\leftrightarrow A_2A_5A_6$，

$a_{17}\leftrightarrow A_3A_4A_5$，　$a_{18}\leftrightarrow A_3A_4A_6$，　$a_{19}\leftrightarrow A_3A_5A_6$，　$a_{20}\leftrightarrow A_4A_5A_6$，

故　$A_1\longleftrightarrow A\backslash\{a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{10}\}$

$\longleftrightarrow\{a_{11},a_{12},a_{13},a_{14},a_{15},a_{16},a_{17},a_{18},a_{19},a_{20}\}$

用以表达 A_1 必须掌握 $a_{11},a_{12},\cdots,a_{20}$位信息.

$A_2\longleftrightarrow A\backslash\{a_1,a_2,a_3,a_4,a_{11},a_{12},a_{13},a_{14},a_{15},a_{16}\}$

$\longleftrightarrow\{a_5,a_6,a_7,a_8,a_9,a_{10},a_{17},a_{18},a_{19},a_{20}\}$

$A_3\longleftrightarrow\{a_2,a_3,a_4,a_8,a_9,a_{10},a_{14},a_{15},a_{16},a_{20}\}$

$A_4\longleftrightarrow\{a_1,a_3,a_4,a_6,a_7,a_{10},a_{12},a_{13},a_{16},a_{19}\}$

$A_5\longleftrightarrow\{a_1,a_2,a_4,a_5,a_7,a_9,a_{11},a_{13},a_{15},a_{18}\}$

$A_6 \longleftrightarrow \{a_1, a_2, a_3, a_5, a_6, a_8, a_{11}, a_{12}, a_{14}, a_{17}\}$

上面构造过程保证了每 3 位的组合都只缺 1 位数. 设 $A_iA_jA_k$ 对应于 a_h，另增一位 A_l，则 $A_iA_jA_l$，$A_iA_kA_l$ 和 $A_jA_kA_l$ 分别对应于 a_p, a_q, a_r. 显然 $a_p, a_q, a_r \neq a_h$. 这样保证了任意 4 位定能恢复密钥 $a_1a_2\cdots a_{20}$的全体.

(a) 沙米尔提出一种基于拉格朗日插值公式的密钥分存思想. 过$(x_i, y_i), i=1,2,\cdots,k$ 点可有一个 $k-1$ 次多项式

$$
\begin{aligned}
p(x) &= y_1 \frac{(x-x_2)(x-x_3)\cdots(x-x_k)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} \\
&\quad + y_2 \frac{(x-x_1)(x-x_3)\cdots(x-x_k)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} \\
&\quad + \cdots + y_k \frac{(x-x_1)(x-x_2)\cdots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})} \\
&= \sum_{j=1}^{k} y_j \prod_{i\neq j} \frac{(x-x_i)}{(x_j-x_i)}
\end{aligned}
$$

设 $GF(q)$是一有限域，$q>n$. 任意选取 $a_1, a_2, \cdots, a_{k-1} \in GF(q)$. 构造一多项式

$$f(x) = k^* + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1}.$$

其中 k^* 是密钥.

令 α 是 $GF(q)$域的本原元素. 作

$$k_i = f(\alpha^i), \qquad i=1,2,\cdots,n$$

称 k_i 为子密钥，将 k_i 交给合作者 A_i 保管. 设有 k 个合作者 A_{i_1}，A_{i_2}，…，A_{i_k}分别提供了各自的子密钥 k_{i_1}，k_{i_2}，…，k_{i_k}以及各自的序号 i_1，i_2，…，i_k. 利用插值公式

$$p^*(x) = \sum_{j=1}^{k} k_{i_j} \prod_{l\neq j} \frac{(x-\alpha^{i_l})}{(\alpha^{i_j}-\alpha^{i_l})}$$

得一个 $k-1$ 次多项式. 显然

$$p^*(\alpha_{i_j}) = k_{i_j}, \qquad j=1,2,\cdots,k$$

$\therefore \quad p^*(x) = f(x)$

$k^* = f(0)$

所以密钥 k^* 得以恢复. 如若只有 A_{i_1}, A_{i_2}, …, $A_{i_{k-1}}$ 个合作者, 不足以确定 $f(x)$, 因而也不能得到 k.

(b) 阿斯木司一勃龙蒙（Asmuth-Bloom）体系：选一素数 $p>k$，另选一组两两互素的整数 m_1, m_2, …, m_n，并满足和 p 互素的条件，此外还要满足：

设 $m_1<m_2<\cdots<m_n$

$$m_1m_2\cdots m_t>pm_nm_{n-1}\cdots m_{n-t+2}$$

令 $\quad l<\lfloor m_1m_2\cdots m_t/p\rfloor$，作

$$L=k+lp<p+lp=(l+1)p\leqslant m_1m_2\cdots m_t$$

令

$$L\equiv k_i(\mathrm{mod}\ m_i),\qquad 1\leqslant i\leqslant n$$

k_i 作为子密钥分别交 A_i 保管. 可以证明若存在 t 个 k_i 便可恢复 k. 这是根据中国剩余定理

$$L\equiv k_{i_j}\ (\mathrm{mod}\ m_{i_j}),\quad i=1,2,\cdots,t$$

模 $m_{i_1}m_{i_2}\cdots m_{i_t}$ 有唯一解. 因：

$$L\leqslant m_1m_2\cdots m_t\leqslant m_{i_1}m_{i_2}\cdots m_{i_t}$$

$$k=L-lp$$

若是只有 $k_{i_1}, k_{i_2}, \cdots, k_{i_{t-1}}$ 这 $t-1$ 个子密钥，模 $m_{i_1}m_{i_2}\cdots m_{i_{t-1}}$ 有唯一解 M，使 $L\equiv M\ (\mathrm{mod}\ m_{i_1}m_{i_2}\cdots m_{i_{t-1}})$

令 $\quad L=M+am_{i_1}m_{i_2}\cdots m_{i_{t-1}}$

这里 $\quad 0\leqslant a<m_1m_2\cdots m_t/(m_{i_1}m_{i_2}\cdots m_{i_{t-1}})$

由于 $\quad m_1m_2\cdots m_t/(m_nm_{n-1}\cdots m_{n-t+2})>p$

所以 a 的值可以是小于 p 的正整数. 由于 $(m_i, p)=1$, $i=1$, 2, …, n. 所以 $(m_{i_1}m_{i_2}\cdots m_{i_{t-1}}, p)=1$. 随着 a 的值在 $0\leqslant a<p$ 区间变化，$L\ (\mathrm{mod}\ p)$ 的值也在这范围内变化，即 L 的值不

能确定.

McEliece 和 Sawate 将 Reed-Solomon 码用在密钥分存技术上，以防止个别合作者的子密钥被篡改或有意提供假的 k_i. 它的基本思想是将密钥 k 利用 Reed-Solomon 码译成码文，分别由 n 个用户保存一部份，根据 Reed-Solomon 码具有纠错的功能，对若干（比如 t 个）错误予以恢复，以达到上述目的.

§ 21　零知识证明问题

（1）协议.

密码协议是一系列的约定，通信系统中的成员都必须遵守以建立系统内部的信息交换和联系. 例如下面证明身份的协议.

每一用户，设为 A，秘密选择一正整数 a 并计算

$$y_A=\alpha^a$$

并用 (A, y_A) 形式公布在通信录上，其中 A 为用户的身份，假定每一用户都有一本通信录.

若 A 要向 B 证明自己的身份，步骤如下：

(a) A 向 B 送去一正整数 α.

(b) B 收到 α 后随机选择一正整数 R，计算

$$y_2=\alpha^R$$

并将 y_2 送给 A.

(c) A 计算

$$y_3=(y_2)^a=\alpha^{aR}$$

并送 y_3 给 B.

(d) B 计算

$$(y_A)^R=\alpha^{aR}$$

并检查 $(y_A)^R = y_3$? 若等号成立，A 的身份便得到了证明.

又例，著名的沙米尔协议也是饶有趣味的. 假定用户 i 有自己的加密算法 E_i 和相应的解密算法 D_i. 假定 A 要向 B 送去明文 m，步骤如下：

(a) A 向 B 送去 $D_A(m)$；

(b) B 退给 A 以 $D_B(D_A(m))$；

(c) A 再送给 B 以

$$E_A(D_B(D_A(m))) = E_A(D_A(D_B(m))) = D_B(m)$$

(d) B 作 $E_B(D_B(m)) = m$

这里假定　$D_i(D_j(m)) = D_j(D_i(m))$

沙米尔的协议妙在 A 和 B 间的秘密通信没有私下约定的密钥，通过 3 次来回便将 m 秘密地送给了 B. 现将这过程形象地表示如图 2.21.1.

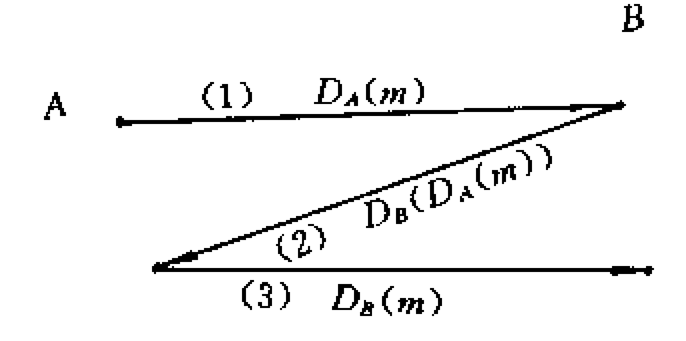


图 2.21.1

(2) 零知识证明是一交互信息交换过程. 一方是证明者 P(prover)，P 拥有无限的计算能力；另一方是验证者 V(verifier)，V 是概率多项式时间机器. 所谓零知识证明说的是证明者 P 要向验证者 V 证明一个事实，若事实是真的，则 V 以很大的概率接受它，如若不然，则 V 也将以很大的概率予以拒绝. 全部过程在多项式时间内完成. 而且验证者并没有从证明

者那里获得任何额外信息，验证者 V 可以独立取得的信息除外，即便 V 采取欺骗手段也仍如此.

(1) 中的协议就不是零知识的. 因为若 B 送去的 $y_2=R$，则他将获得 $y_3=R^a$. 这个结果就不是他所能得到的，虽然这距离他想知道的 a 还很远.

零知识证明举例. 若已知正整数 x 和 y，$0<y<x$，$\gcd(x, y)=1$. P 宣称他知道不存在 $z\in \mathbf{Z}$，使得

$$z^2\equiv y(\bmod x)$$

验证者 V 便产生一组整数序列

$$z_1, z_2, \cdots, z_n$$

及一组 0—1 序列

$$b_1, b_2, \cdots, b_n$$

满足 $0<z_i<x$，$\gcd(z_i, x)=1$，$i=1, 2, \cdots, n$. 并计算

$$w_i=\begin{cases} z_i^2 & (\bmod x) \quad 若 b_i=0 \\ z_i^2 y & (\bmod x) \quad 若 b_i=1 \end{cases} \quad i=1,2,\cdots,n$$

V 将 $w_1, w_2, \cdots, w_n$ 送给 P.

P 对此进行判断，令

$$c_i=\begin{cases} 0, & 若存在 z_i， 满足 z_i^2\equiv w_i(\bmod x), \\ 1, & 其它. \end{cases}$$

P 将 $c_1, c_2, \cdots, c_n$ 送给 V.

若对所有的 i 有 $c_i=b_i$ 成立，则 P 的断言，即不存在 z，使得 $z^2\equiv y(\bmod x)$便得到证明.

表面上 V 得到的回答都是他自己已知道的事实，但实际上 P 已向 V 泄露了 V 本不掌握的信息. 零知识证明的讨论方兴未艾，有很大的理论意义，关系到网络通信的安全问题，可能还不仅限于这点上.

(3) 关于平方剩余的讨论已见于第 2 章 § 3. 假定 p 是素数

(非偶数)，β 是 $GF(p)$域上非零元素，根据费尔玛定理

$$\beta^{p-1}\equiv 1(\bmod\ p)$$

但

$$\beta^{p-1}-1\equiv(\beta^{(p-1)/2}-1)(\beta^{(p-1)/2}+1)(\bmod\ p)$$

如若 $\beta\in QR_p$，即存在 $\alpha\in GF(p)$满足

$$\alpha^2\equiv\beta(\bmod\ p)$$

所以

$$\beta^{(p-1)/2}\equiv\alpha^{p-1}\equiv 1(\bmod\ p)$$

反过来余下的（$p-1$）/2 个非零元素为非平方剩余，它们都满足费尔玛定理，但

$$\beta^{(p-1)/2}\not\equiv 1(\bmod\ p)$$

故

$$\beta^{(p-1)/2}\equiv -1(\bmod\ p)$$

这就证明了下面的结论：

$$\beta^{(p-1)/2}\equiv\begin{cases}1 & \beta\in QR_p\\ -1 & \beta\notin QR_p\end{cases}$$

这是判断 β 是否 mod p 平方剩余的方法.

§ 22　序列密码和线性反馈移位寄存器(LFSR)

（1）序列密码.

商农提出完全安全的密码概念.由此引出一次一密的构想，若能以某种方式产生一随机序列，利用这样的序列对明文进行加密，如图 2.22.1 所示.

即设密钥流 $k=k_1k_2\cdots$，明文 $m=m_1m_2\cdots$和密文 $c=c_1c_2\cdots$，都是 0—1 序列

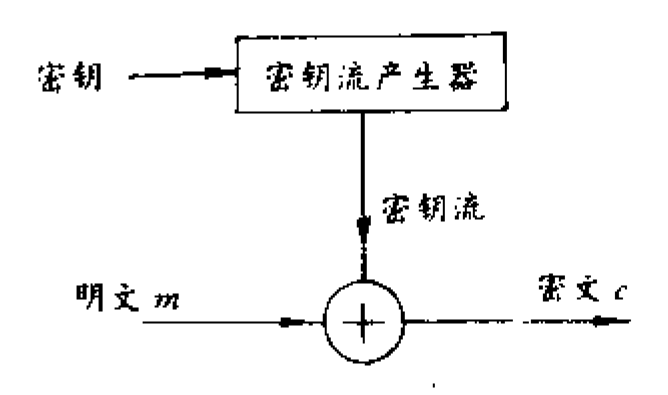


图 2.22.1

$$c_i \equiv m_i + k_i \pmod 2), \qquad i=1, 2, \cdots$$

用 $c_i = m_i \oplus k_i$ 表示它.

在这一节主要讨论用线性反馈移位寄存器产生一种称之为 m 的序列. 用序列进行加密的方法叫做序列密码.

(2) 线性反馈移位寄存器，它可用图 2.22.2 来表示，这样的线性反馈移位寄存器为 n 级的. 线性反馈移位寄存器简记为 LFSR.

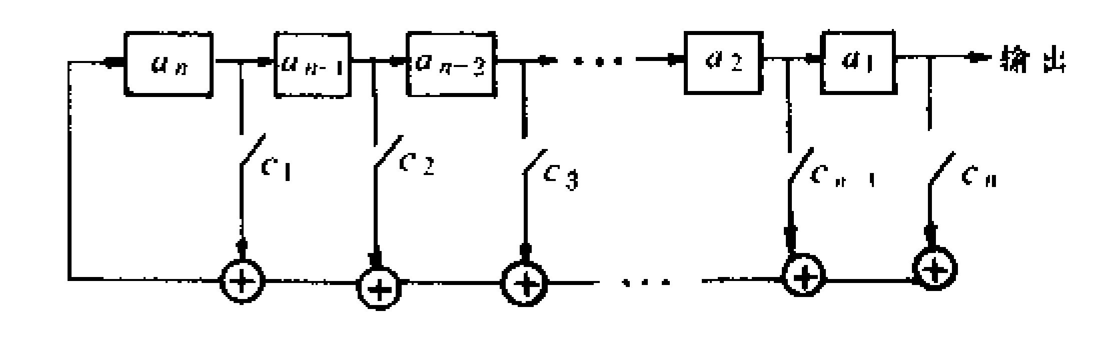


图 2.22.2

其中 $\qquad a_i(t+1) = a_{i+1}(t), \qquad i=1,2,\cdots,n-1$

$$a_n(t+1) = c_n a_1(t) + c_{n-1} a_2(t) + \cdots + c_2 a_{n-1}(t) + c_1 a_n(t)$$

其中 $+$ 是 $\oplus$ 的简写. 开关 c_i 的断开和闭合分别表示 $c_i=0$ 或 1.

一般都假定 $c_n=1$.

如图 2.22.3 所示，

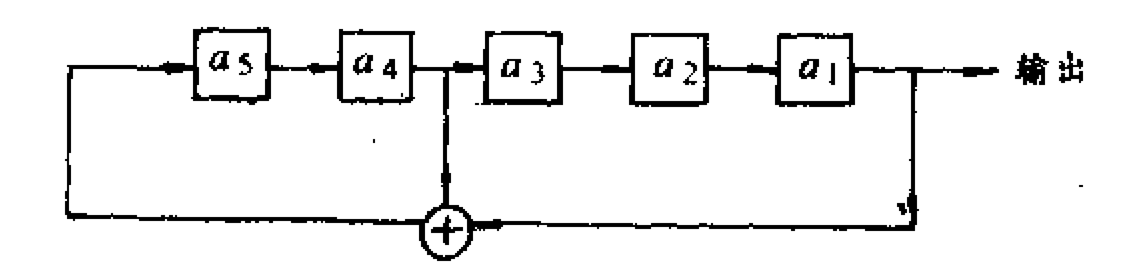


图 2.22.3

它表达

$$a_i(t+1)=a_{i+1}(t),\qquad i=1,2,3,4$$

$$a_5(t+1)=a_1(t)+a_4(t)$$

如若 $t=0$ 时，$a_1=1$，$a_2=0$，$a_3=1$，$a_4=a_5=0$；

$t=1$ 时应为 $a_1=0$，$a_2=1$，$a_3=a_4=0$，$a_5=a_1+a_4=1$，输出为 $a_1=1$.

$t=2$ 时应有 $a_1=1$，$a_2=a_3=0$，$a_4=1$，$a_5=0$，输出为 1.

t	a_5	a_4	a_3	a_2	a_1	t	a_5	a_4	a_3	a_2	a_1
0	0	0	1	0	1	19	1	0	0	0	1
1	1	0	0	1	0	20	1	1	0	0	0
2	0	1	0	0	1	21	1	1	1	0	0
3	0	0	1	0	0	22	1	1	1	1	0
4	0	0	0	1	0	23	1	1	1	1	1
5	0	0	0	0	1	24	0	1	1	1	1
6	1	0	0	0	0	25	0	0	1	1	1
7	0	1	0	0	0	26	1	0	0	1	1

t	a_5	a_4	a_3	a_2	a_1	t	a_5	a_4	a_3	a_2	a_1
8	1	0	1	0	0	27	1	1	0	0	1
9	0	1	0	1	0	28	0	1	1	0	0
10	1	0	1	0	1	29	1	0	1	1	0
11	1	1	0	1	0	30	0	1	0	1	1
12	1	1	1	0	1	31	0	0	1	0	1
13	0	1	1	1	0	32	1	0	0	1	0
14	1	0	1	1	1	33	0	1	0	0	1
15	1	1	0	1	1	34	0	0	1	0	0
16	0	1	1	0	1	35	0	0	0	1	0
17	0	0	1	1	0	36	0	0	0	0	1
18	0	0	0	1	1						

这个线性反馈移位寄存器的输出为序列

1010010000101011101100011111001101001…

不难想像 LFSR 的输出序列将是周期的，因为一当寄存器上的状态出现重复，则以后的状态周而复始. n 级的 LFSR 最多只有 2^n-1 种非零状态，重复是不可避免的. 而且周期 $r\leqslant 2^n-1$. 若 n 个寄存器的状态为全零，则这 LFSR 将一直保持为全零状态.

(3) 特征多项式.

n 级的 LFSR 输出的递推序列 $\{a_i\}$ 满足当 $j\geqslant n$ 时有

$$a_{j+1}=c_1a_j+c_2a_{j-1}+\cdots+c_{n-1}a_{j-n+2}+c_na_{j-n+1}$$

$$a_1=b_1,\quad a_2=b_2,\quad \cdots,\quad a_n=b_n$$

称 $p(x)=1+c_1x+c_2x^2+\cdots+c_nx^n$ 为对应的特征多项式. 对于序列 $\{a_i\}$ 有母函数

$$G(x)=a_1+a_2x+a_3x^2+\cdots=\sum_{i=1}^{\infty}a_ix^{i-1}$$

定理 2.22.1 $G(x)=\varphi(x)/p_n(x)$，其中

$$\varphi(x)=\sum_{i=1}^{n}c_{n-i}x^{n-i}\sum_{j=1}^{i}a_jx^{j-1}$$

$$p_n(x)=\sum_{i=0}^{n}c_ix^i,\qquad c_0=1$$

证：$a_{n+1}=c_1a_n+c_2a_{n-1}+\cdots+c_na_1$

$a_{n+2}=c_1a_{n+1}+c_2a_n+\cdots+c_na_2$

上面等式等号两端分别乘以 x^n，x^{n+1}，…并求和得

$$\begin{aligned}&G(x)-(a_1+a_2x+\cdots+a_nx^{n-1})\\&=c_1x[G(x)-(a_1+a_2x+\cdots+a_{n-1}x^{n-2})]\\&+c_2x^2[G(x)-(a_1+a_2x+\cdots+a_{n-2}x^{n-3})]\\&+\cdots\\&+c_nx^nG(x)\end{aligned}$$

移项整理后可得

$$\begin{aligned}&(1+c_1x+c_2x^2+\cdots+c_nx^n)G(x)\\&=(a_1+a_2x+\cdots+a_nx^{n-1})\\&\quad+c_1x(a_1+a_2x+\cdots+a_{n-1}x^{n-2})\\&\quad+c_2x^2(a_1+a_2x+\cdots+a_{n-2}x^{n-3})\\&\quad+\cdots\\&\quad+c_{n-1}x^{n-1}a_1\end{aligned}$$

即

$$p_n(x)G(x)=\sum_{i=1}^{n}c_{n-i}x^{n-i}\sum_{j=1}^{i}a_jx^{j-1}=\varphi(x)$$

注意 $a+a=0$. 定理得证.

定理 2.22.2 $p_n(x)$是序列$\{a_i\}$的特征多项式，且 $p_n(x)|x^m+1$，则$\{a_i\}$的周期 $r|m$.

证：设存在 $q(x)$，使得

$$p_n(x)q(x) = x^m + 1$$

但 $$p_n(x)G(x) = \varphi(x)$$

$$\therefore \quad p_n(x)q(x)G(x) = q(x)\varphi(x)$$

即 $$(x^m + 1)G(x) = q(x)\varphi(x)$$

$q(x)$的方次为 $m-n$，$\varphi(x)$的方次不超过 $n-1$，故 $(x^m+1)G(x)$ 的方次不超过$(m-n)+(n-1)=m-1$. 这就证明了

$$a_{i+m}=a_i$$

对于任意正整数 i 都成立. 设 $m=kr+t$，$0\leqslant t<r$，

$$a_{i+m}=a_{i+kr+t}=a_{i+t}=a_i$$

$$\therefore \quad t=0$$

即 $$r \mid m$$

(4) n 级 LFSR 的输出序列的周期 r 不依赖于初始条件，而依赖于特征多项式 $p_n(x)$. 我们感兴趣的是 LFSR 遍历 2^n-1 个非零状态，这时序列的周期达到最大，即 2^n-1，这样的序列称为 m 序列. 显然对于特征多项式一样，而仅仅初始条件不同的两个 LFSR 输出的序列，一个设为$\{a_i^{(1)}\}$，另一个是$\{a_i^{(2)}\}$，则其中一个必是另一个的移位，也就是存在一个常数 k，使得

$$a_i^{(1)} = a_{k+i}^{(2)}, \qquad i = 1,2,\cdots$$

但序列$\{a_i\}$和母函数一一对应，故$S(p_n(x))$也就是以$p_n(x)$为特征多项式的母函数集合.

定理 2.22.3 设$\{a_i\} \in S(p(x))$，$\{b_i\} \in S(q(x))$，则

$$\{a_i + b_i\} \in S(r(x))$$

其中 $r(x) = \mathrm{lcm}\{p(x), q(x)\}$

证： 设$r(x) = a(x)p(x) = b(x)q(x)$

$\{a_i\}$的母函数为$A(x)$，$\{b_i\}$的母函数为$B(x)$，

$$A(x) = \alpha(x)/p(x), \qquad B(x) = \beta(x)/q(x)$$

$\alpha(x)$的次方$<p(x)$的方次，$\beta(x)$的方次$<q(x)$的方次.$\{a_i + b_i\}$的母函数为

$$A(x) + B(x) = \frac{\alpha(x)}{p(x)} + \frac{\beta(x)}{q(x)}$$

$$= \frac{a(x)\alpha(x) + b(x)\beta(x)}{r(x)}$$

$a(x)\alpha(x)$和$b(x)\beta(x)$的方次均小于$r(x)$的方次

$\therefore \quad A(x) + B(x) \in S(r(x))$

定理 2.22.4 若(1) $p(x)$是n次多项式，

(2) $m = \min\{k \mid p(x) \mid x^k + 1\}$

(3) $\{a_i\} \in S(p(x))$，则$\{a_i\}$的周期$r \mid m$.

证： 设$x^m + 1 = p(x)q(x)$，$\{a_i\}$序列的母函数

$$G(x) = \varphi(x)/p(x) = \varphi(x)q(x)/(1 + x^m)$$

$$= \varphi(x)q(x)[1 + x^m + x^{2m} + \cdots]$$

其中$q(x)$的方次为$m - n$.所以$\varphi(x)q(x)$的方次$<m$.这就证明了序列$\{a_i\}$的周期r除尽m，即$r \mid m$.

定理 2.22.5 若(1) $p(x)$是n次不可化约多项式；

(2) $m = \min\{k \mid p(x) \mid x^k + 1\}$；

(3) $\{a_i\} \in S(p(x))$；

则序列$\{a_i\}$的周期为m.

证：设序列 $\{a_i\}$ 的周期为 r，则根据上面的定理 2.22.4，$r|m$. 故它的母函数为

$$G(x) = h(x)/(1 + x^r)$$

$h(x)$ 的方次 $< r$. 但

$$G(x) = \varphi(x)/p(x)$$

$$\therefore \quad \varphi(x)/p(x) = h(x)/(1 + x^r)$$

但 $p(x)$ 是不可化约的，面且 $\varphi(x)$ 的方次 $< n$，这就证明了 $p(x)|(1+x^r)$. 但 $p(x)|(1+x^m)$，而且 $m = \min\{k|p(x)|(1+x^k)\}$，故 $\quad r = m$.

定理 2.22.6 若(1) $p(x)$ 是 n 次多项式；(2) $\{a_i\} \in S(p(x))$ 的周期为 $2^n - 1$，则 $p(x)$ 是不可化约的.

证：如若不然，令 $p(x) = q(x)r(x)$，其中 $q(x)$ 为 k 次不可化约的多项式，则

$$\frac{1}{q(x)} \in S(q(x))$$

而且 $1/q(x)$ 的周期 $r_1|2^k - 1$.

另一方面

$$\frac{1}{q(x)} = \frac{r(x)}{p(x)} \in S(p(x))$$

这说明以 $1/q(x)$ 为母函数的序列是 $\{a_i\}$ 的某一种位移. 它的周期是 $2^n - 1$，这只有一种可能 $n = k$，即

$$p(x) = q(x).$$

综上可得证. 证毕.

定理 2.22.7 以 n 次多项式 $p(x)$ 为特征多项式的 n 级 LFSR 所输出的 $\{a_i\}$ 是 m 序列(即周期为 $2^n - 1$) 的充要条件是

(1) $p(x)$ 是不可化约的；

(2) $\min\{k|p(x)|x^k + 1\} = 2^n - 1$.

证：必要性.

因$\{a_i\} \in S(p(x))$,根据定理2.22.6,$p(x)$是不可化约的.又根据定理2.22.4,$\{a_i\}$的周期2^n-1必然除尽$p(x)$的周期m,

$$m = \min\{k \mid p(x) \mid x^k + 1\}$$

但有限域理论告诉我们,n次多项式$p(x)$应有

$$p(x) \mid x^{2^n-1} + 1$$

即 $$m = 2^n - 1$$

充分性.

已知$p(x)$是n次不可化约多项式，而且周期为2^n-1. 从定理2.22.5可知$\{a_i\}$的周期也是2^n-1,即$\{a_i\}$是m序列.

§23　m序列的若干性质

为了讨论方便先介绍几个概念. 比如序列

0011101001111

开始出现00，我们称之为0的2游程，接着111. 称之为1的3游程. 其余依此类推.

设 $\{a_i\}$ 是以r为周期的序列. 取

$$a_1,\ a_2,\ \cdots,\ a_r$$

$$a_{1+\tau},\ a_{2+\tau},\ \cdots,\ a_{r+\tau}$$

并逐次比较，计统计$a_i=a_{i+\tau}$的总数，令之为n_τ; $a_j \neq a_{j+\tau}$的数目令之为d_τ，定义

$$R_\tau = \frac{n_\tau - d_\tau}{r}$$

下面将讨论n级的LFSR，若输出的序列$\{a_i\}$的周期达到2^n-1，则具有如下的特性：

(a) 若 r 是奇数，则 $\{a_i\}$ 在一周期内 0 的个数比 1 的个数多 1 个或少 1 个. 若 r 是偶数时，则 0 的个数等于 1 的个数.

(b) 1 游程的数目为游程总数的 1/2，2 游程的数目占游程总数的 $1/2^2$，…，c 游程数目占游程总数的 $1/2^c$. $c\neq n, n-1$ 时，0 的 c 游程数目和 1 的 c 游程数目应相等.

(c) R_τ 是一个常数.

证 由于 n 级的 LFSR 的状态遍历 2^n-1 个非零的 n 位 2 进制数. 假设 n 位 2 进制数用 $a_1a_2\cdots a_n$ 来表示，a_1 位有 2^{n-1}个是 1，$2^{n-1}-1$ 个是零，这就证明了(a) 的结论.

由于 LFSR 不会出现全零状态，故没有 0 的 n 游程. 有 1 个且仅有 1 个 1 的 n 游程. 不可能出现 1 的 $n+1$ 游程. 如若不然将出现两个全 1 的状态相邻，则整个序列是全 1. 这是不允许的. 故必然出现

$$0\underbrace{1\cdots1}_{n\text{位}}0$$

一段的子序列，故 LFSR 的状态依次为

$$0\underbrace{1\cdots1}_{n\text{位}},\quad \underbrace{1\cdots1}_{n\text{位}},\ \underbrace{1\cdots1}_{n\text{位}}0$$

由于$0\underbrace{1\cdots1}_{n\text{位}}$和$\underbrace{1\cdots1}_{n\text{位}}0$只能出现 1 次. 因此不会出现 1 的 $n-1$ 游程，即不会出现

$$0\underbrace{1\cdots1}_{n-1\text{位}}0$$

子序列. 但会出现

$$1\underbrace{0\cdots0}_{n-1\text{位}}1$$

子序列. 它产生 $10\cdots0$ 和 $0\cdots01$ 两种状态.

对于 $n>2$，c 为不超过 $n-2$ 的任一正整数，则 1 的 c 游程

的数目可以从考察 LFSR 状态为

$$\underbrace{01\cdots10}_{c+2\text{位}}\ \underbrace{\cdots\cdots}_{n-c-2\text{位}}$$

时得出.

这样的状态共有 2^{n-c-2} 个. 类似的理由可以制定 0 的 c 游程数目也应是 2^{n-c-2}. 所以在一个循环周期内 1 的游程数目应为

$$1+\sum_{c=1}^{n-2}2^{n-c-2}=1+1+2^1+2^2+\cdots+2^{n-3}=2^{n-2}$$

0 的游程数也是 2^{n-2}. (b) 的结论正确.

由于序列满足递推关系

$$a_{k+n}=c_1a_{k+n-1}+c_2a_{k+n-2}+\cdots+c_na_k$$

所以

$$a_{k+n+\tau}=c_1a_{k+n+\tau-1}+c_2a_{k+n+\tau-2}+\cdots+c_na_{k+\tau}$$

$$\begin{aligned}\therefore\quad a_{k+n}+a_{k+n+\tau}=&c_1(a_{k+n-1}+a_{k+n+\tau-1})\\&+c_2(a_{k+n-2}+a_{k+n+\tau-2})\\&+\cdots+c_n(a_k+a_{k+\tau})\end{aligned}$$

令 $\qquad b_k=a_k+a_{k+\tau}$

则 $\qquad b_{k+n}=c_1b_{k+n-1}+c_2b_{k+n-2}+\cdots+c_nb_k$

也就是序列 $\{a_i+a_{i+\tau}\}=\{b_i\}$,也和 $\{a_i\}$ 一样满足相同的递推关系. 所以问题导致讨论序列 $\{b_i\}$ 中 0 的个数和 1 的个数. 这个差数为 -1,故.

$$R_\tau=(2^{n-1}-1-2^{n-1})/2^{n-1}=-1/(2^n-1)$$

(c)的正确性获得证明.

§24 非线性的反馈移位寄存器

(1) 我们可将线性反馈的移位寄存器的概念予以拓广, 把

它看作是一种自动机，且是由三部分构成的.

(a) 有限状态集 $S=\{S_i|i=1,2,\cdots,l\}$.

(b) 有限输入字符集

$$A_1=\{a_j^{(1)}|j=1,2,\cdots,m_1\}$$

和有限输出字符集

$$A_2=\{a_k^{(2)}|k=1,2,\cdots,m_2\}$$

(c) 转移变换.

它具有这样的一种功能，当自动机处于 S_i 状态时，输入 $A_j^{(1)}$ 后，状态改变为 S_k，并输出一字符 $A_k^{(2)}$. 即

$$A_k^{(2)}=f_1(A_j^{(1)},S_i),\qquad S_k=f_2(A_j^{(1)},S_i)$$

例 1 $S=\{S_1,S_2,S_3\}$, $A_1=\{a_1^{(1)},a_2^{(1)},a_3^{(1)}\}$

$A_2=\{a_1^{(2)},a_2^{(2)},a_3^{(2)}\}$

f_1:	$a_1^{(1)}$	$a_2^{(1)}$	$a_3^{(1)}$
S_1	$a_1^{(2)}$	$a_3^{(2)}$	$a_2^{(2)}$
S_2	$a_2^{(2)}$	$a_1^{(2)}$	$a_3^{(2)}$
S_3	$a_3^{(2)}$	$a_2^{(2)}$	$a_1^{(2)}$

f_2:	$a_1^{(1)}$	$a_2^{(1)}$	$a_3^{(1)}$
S_1	S_2	S_1	S_3
S_2	S_3	S_2	S_1
S_3	S_1	S_3	S_2

上面这个自动机可用有限图表示于图 2.24.1.

每一顶点表示一种状态. 从状态 S_i 到 S_j 的有向边表示状态的转移. 边上的字符串($a_i^{(1)}$ 时 $a_j^{(2)}$)表明在状态 S_i 下输入字符 $a_i^{(1)}$

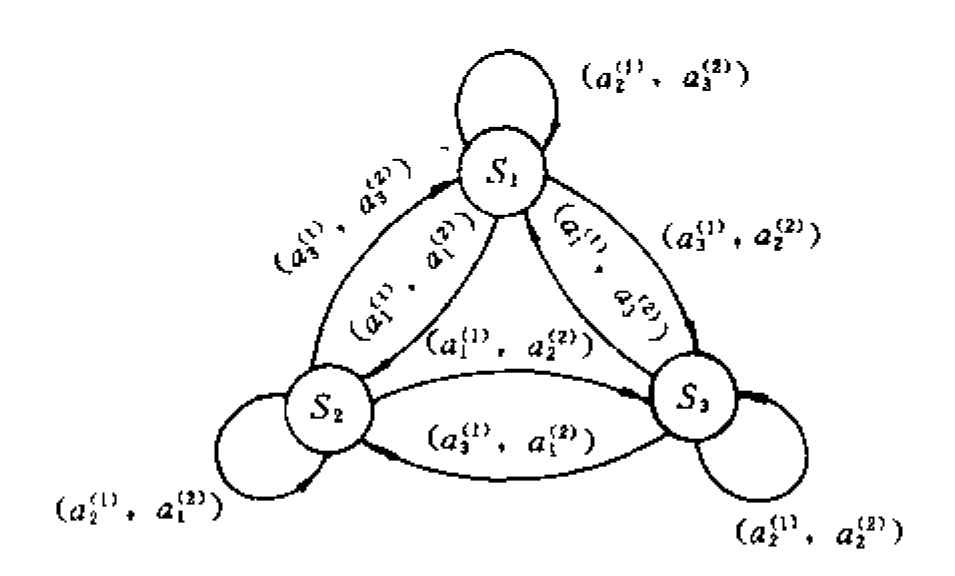


图 2.24.1

时，输出的字符是 $a_j^{(2)}$，并将状态改为 S_j.

(2) 一般的移位寄存器可用图 2.24.2 来表达.

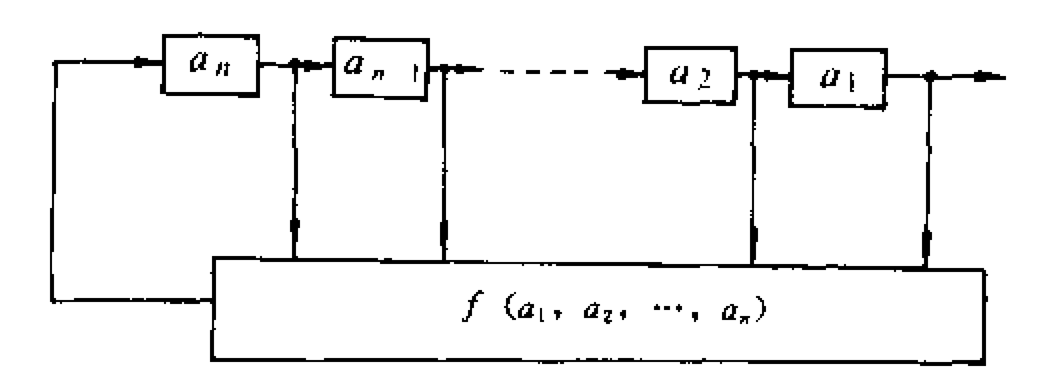


图 2.24.2

其中函数 $f(a_1, a_2, \cdots a_n)$ 可以用一真值表来表示. 这样的布尔函数有 2^{2^n} 个.

例 2 德布鲁恩 (de Bruijn) 序列

图 2.24.3 有 8 个顶点编号从 000 到 111. 从 $(a_1a_2a_3)$ 到 $(a_2a_3a_4)$ 有一条有向边 $a_1a_2a_3a_4$，表达从 $(a_1a_2a_3)$ 状态转向状态 $(a_2a_3a_4)$，而且每一个顶点 (a_1, a_2, a_3) 有两条有向边

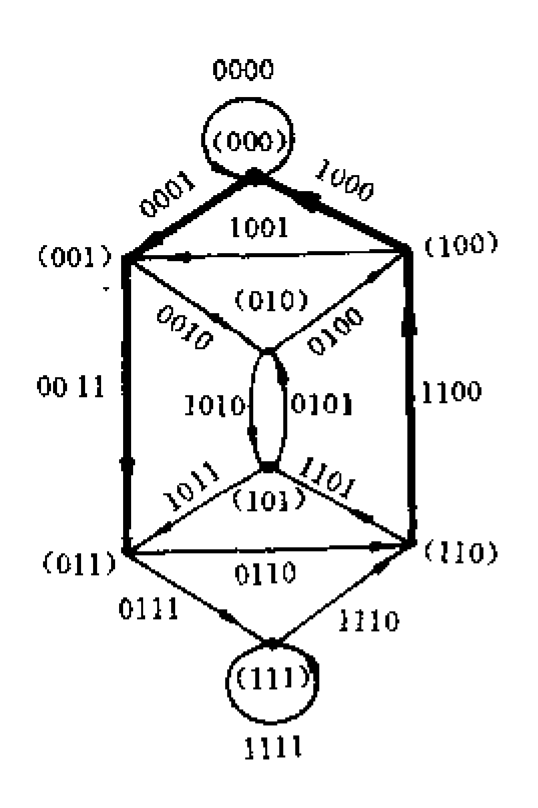


图 2.24.3

$(a_1a_2a_3 0)$ 和 $(a_1a_2a_3 1)$

表达在状态 $(a_1a_2a_3)$ 时输入为 0 或 1 时状态转移图．同样每一个顶点 $(a_1,\ a_2,\ a_3)$ 也有两条边进入，一为 $(0a_1a_2a_3)$，另一是 $(1a_1a_2a_3)$．

图 2.24.3 是 $n=3$ 的 de Bruijn 图．一般说来 n 级移位寄存器的状态转换图都是德布鲁恩图的子图．

由 n 级移位寄存器产生的周期达到 2^n 的序列叫做 M 序列．

（3）下面介绍若干个由 LFSR 构成的非线性序列

（a）如图 2.24.4 所示，LFSR1 和 LFSR2 分别是 m 级和 n 级线性移位寄存器，J—K 触发器的功能见表 2.24.1.

表 2.24.1

J	K	C_k
0	0	c_{k-1}
0	1	0
1	0	1
1	1	$\bar{c}_{k-1}$

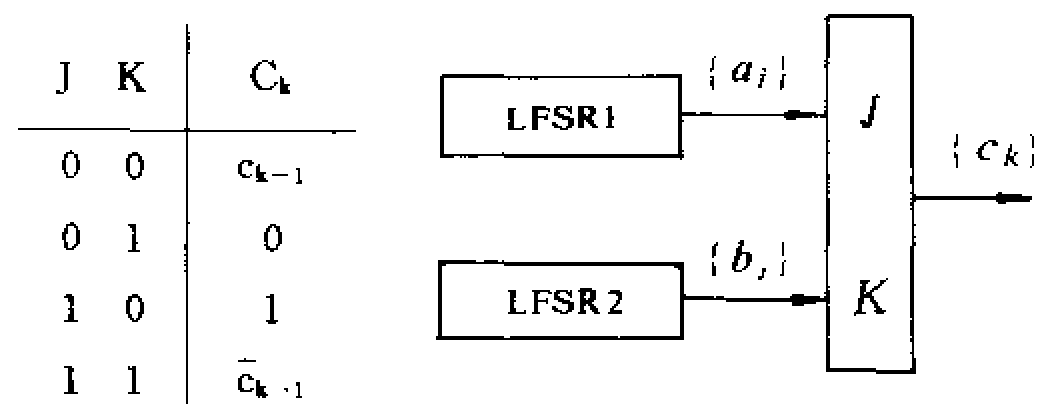


图 2.24.4

令 $c_{-1}=0$，则

$$c_k=(c_k+1)(b_k+1)c_{k-1}+(a_k+1)b_k\cdot 0$$
$$+a_kb_k(c_{k-1}+1)+c_k(b_k+1). \qquad (2.24.1)$$

即表 2.24.1 的功能可用表达式（2.24.1）表达．故

$c_0 = a_0$

$c_1 = (a_1 + b_1 + 1)a_0 + a_1$

$c_2 = (a_2 + b_2 + 1)[(a_1 + b_1 + 1)a_0 + a_1] + a_2$

……

(b) 蒲礼士 (Pless) 利用 (a) 的 J—K 触发器构造一种伪随机序列产生器，如图 2.24.5 所表示的那样. 由 8 个 LFSR 组合而成的.

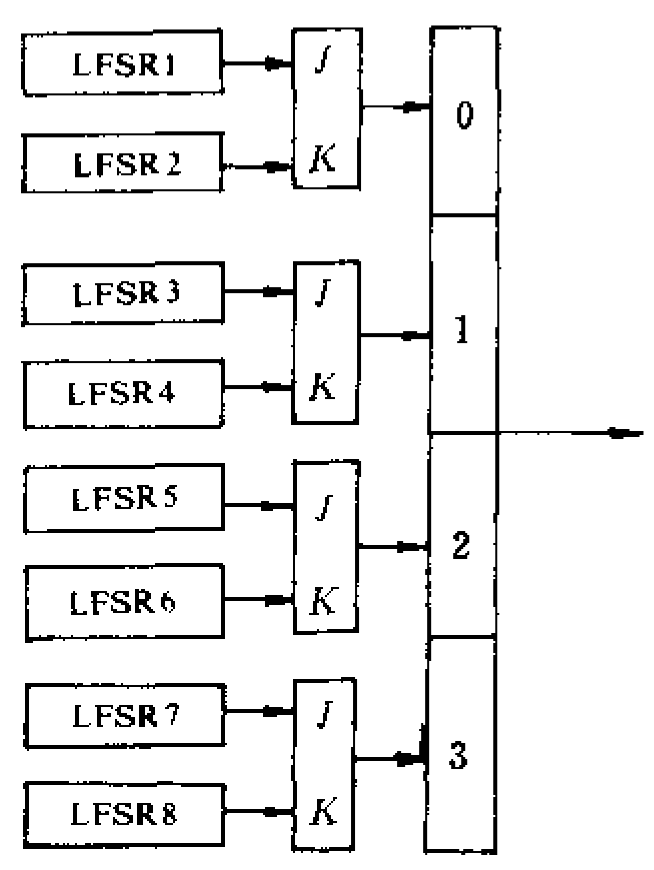


图 2.24.5

这个体制的密钥包含有 8 个 LFSR 的构造和它们的初始状态，以及输出单元的顺序.

(c) 如图 2.24.6 表示的那样，由 LFSR1 的状态作为地址到 LFSR2 中取该地址单元的值. 例如 LFSR1 的状态为 $a_1a_2a_3$，即为 3 级线性移位寄存器，LFSR2 为 8 级线性移位寄存器，状态

为 $b_1b_2b_3b_4b_5b_6b_7b_8$. 例如，若 $(a_1a_2a_3)=(0\ 1\ 1)$，则输出的 c 应为 b_3.

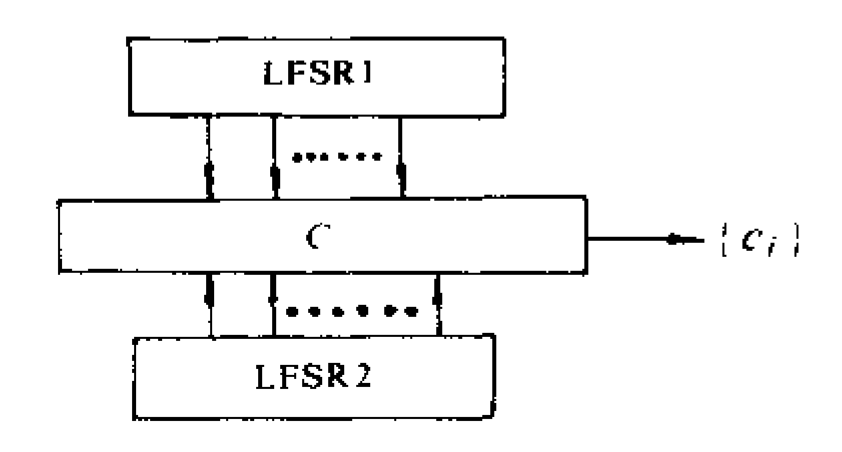


图 2-24-6

也可以从 LFSR1 中某几位的状态到 LFSR2 中相应的单元中取值.

例

假如 LFSRA 的初始状态为1 0 0 0,即$t=0$时$(a_3a_2a_1a_0)=(1\ 0\ 0\ 0)$，LFSR$B$ 的初始状态为0 1 0. 输出的是 a_h，其中 $h=b_2b_0$.（如图2.24.7所示）输出序列如下表：

t	a_3	a_2	a_1	a_0	b_2	b_1	b_0	c_i
0	1	0	0	0	0	1	0	0
1	0	1	0	0	1	0	1	0
2	0	0	1	0	1	1	0	0
3	1	0	0	1	1	1	1	1
4	1	1	0	0	0	1	1	0
5	0	1	1	0	0	0	1	1
6	1	0	1	1	1	0	0	0
7	0	1	0	1	0	1	0	1
8	1	0	1	0	1	0	1	1
9	1	1	0	1	1	1	0	1
10	1	1	1	0	1	1	1	1
11	1	1	1	1	0	1	1	1
12	0	1	1	1	0	0	1	1
13	0	0	1	1	1	0	0	0
14	0	0	0	1	0	1	0	1
15	1	0	0	0	1	0	1	1
16	0	1	0	0	1	1	0	1

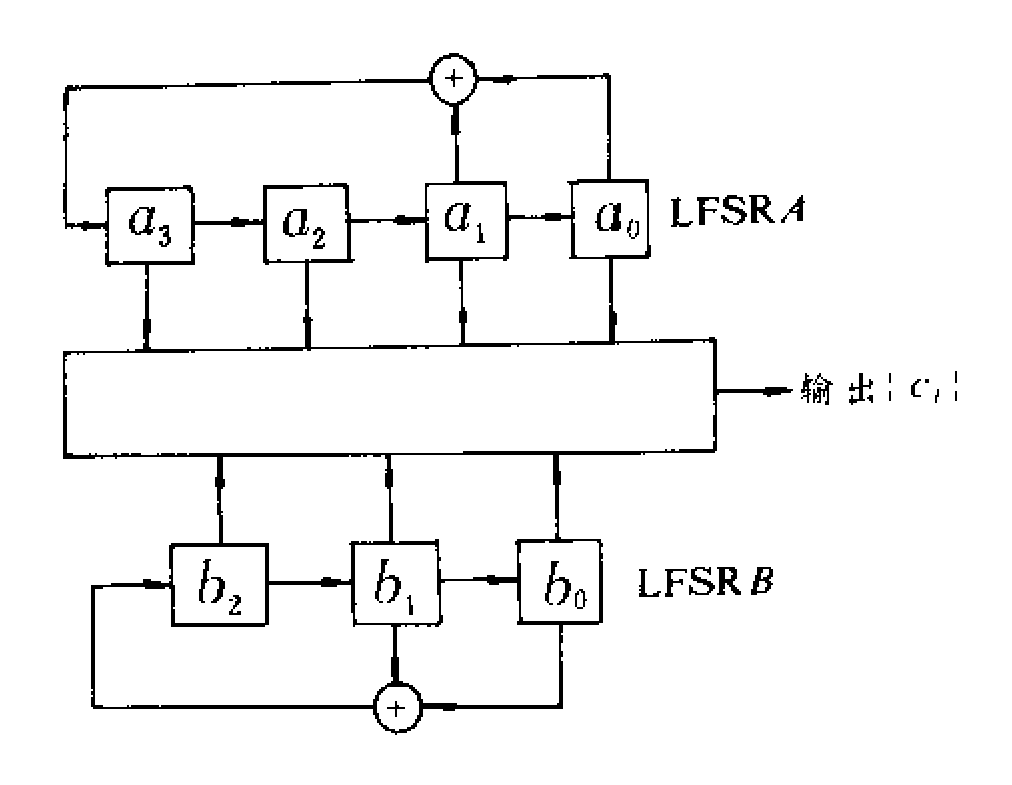


图 2.24.7

(d) 密文反馈加密技术.

利用 LFSR 产生的随机流加密，并用加密后的密文进行反馈的技术如图2.24.8所示.

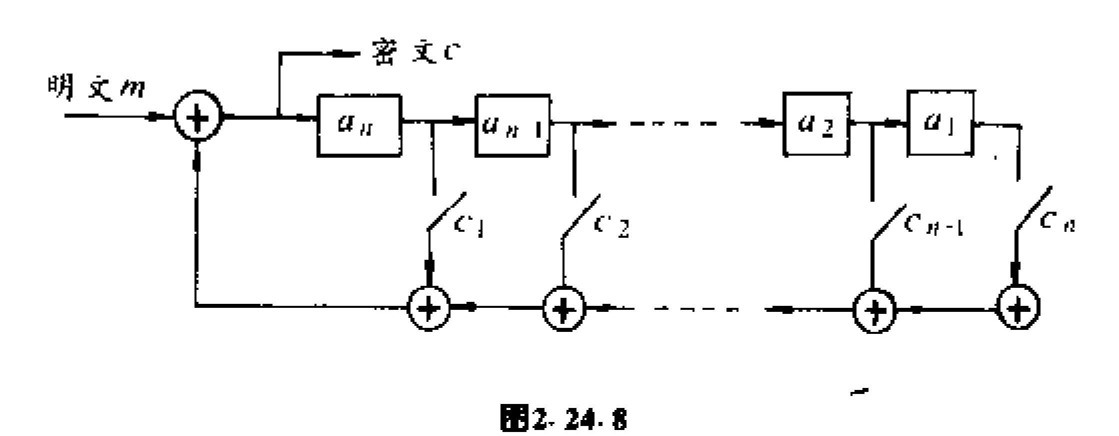


图2.24.8

设明文为

$m=m_1m_2\cdots$

密文 c 为

$$c = e_1 e_2 \cdots$$

则

$$e_1 = m_1 + (c_1 a_n + c_2 a_{n-1} + \cdots + c_{n-1} a_2 + c_n a_1)$$

$$e_2 = m_2 + (c_1 e_1 + c_2 a_n + c_3 a_{n-1} + \cdots + c_n a_2)$$

$$e_3 = m_3 + (c_1 e_2 + c_2 e_1 + c_3 a_n + \cdots + c_n a_3)$$

$$\cdots\cdots$$

$$e_k = m_k + (c_1 e_{k-1} + c_2 e_{k-2} + \cdots c_{k-1} e_1 + c_k a_n + c_{k+1} a_{n-1} + \cdots + c_n a_k), \quad 1 \leqslant k \leqslant n$$

$$\cdots\cdots$$

$$e_{n+1} = m_{n+1} + (c_1 e_n + c_2 e_{n-1} + \cdots + c_n e_1)$$

$$\cdots\cdots$$

$$e_{n+h} = m_{n+h} + (c_1 e_{n+h-1} + c_2 e_{n+h-2} + \cdots + c_n e_h) \quad h \geqslant 1$$

解密过程见图2.24.9.

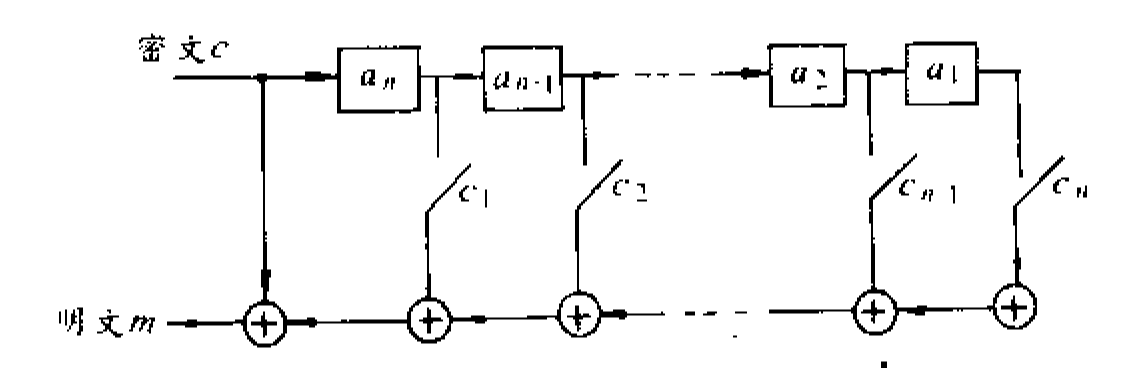


图2.24.9

假如输出的是 $m^* = m_1^* m_2^* \cdots$，则

$$m_1^* = e_1 + (c_1 a_n + c_2 a_{n-1} + \cdots + c_{n-1} a_2 + c_n a_1)$$

将前面(e)中的 e_1 代入便得

$$m_1^* = m_1$$

同样的道理，

$$m_2^* = e_2 + c_1 e_1 + c_2 a_n + c_3 a_{n-1} + \cdots + c_{n-1} a_3 + c_n a_2$$

将(e)中的 e_2 代入便得

$$m_2^* = m_2$$

其它依此类推，可证图2.24.9的解密过程是正确的.

序列密码的密文在传输过程中若丢失或增加一位，则将引起发送方和接收方的伪随机数发生器失去同步，导致解密失败. 但利用上述的密文反馈技术，每一位的 m_{n+k} 只依赖于当前的输入 e_{n+k} 以及 e_{n+k-1}，e_{n+k-2}，…，e_k，所以每一位的差错仅仅对从这一位开始的 n 位有影响，过后影响便消失，一切恢复正常.

$$\cdots \quad \underset{\substack{\uparrow \\ \text{发生差错}}}{e_k} \quad \underbrace{e_{k+1}e_{k+2}\cdots e_{k+n-1}}_{\text{解密失误}} \quad \underset{\substack{\uparrow \\ \text{恢复正常}}}{e_{k+n}} \quad \cdots$$

即当 e_k 从 LFSR 中消失时，它的影响也随之消失.

结　束　语

本书叫做“计算密码学”，不言而喻旨在讨论“近代密码学”．当然，为此也不得不讲一点传统密码．“近代密码”是相对“传统密码”而言，如第一章中见到的那样，传统密码多是通过各种的置换来实现它．近代密码学则需要求助于数学上的计算．从此数学像潮水般涌进了密码学中来，并在近代密码学的研究中扮演了日益重要的角色．数学的许多分支都在这园地里占有自己的位置．读者已经看到，从数论、信息论、概率统计，到群论、有限域理论、组合论、算法复杂性理论、编码理论、自动机理论，以及代数几何中的椭圆曲线等，都对密码学的发展作出了引人入胜的贡献．从这意义上讲，似乎密码学本身就是一门应用数学．本书在涉及如此多的数学问题后，就此打住，作者还想说几句并非多余的话作为结束．那就是：密码学毕竟不就是数学，它有自己的空间．它的产生是时代的需求，是计算机蓬勃发展刺激的结果．它有许多事情要做，是一块待开垦的处女地．密码不仅仅用于保密通信，还是计算机网络、计算机安全、软件保护等学科的基础．这样的理解才比较全面些．

近代密码学奠基于70年代中期，到现在不过十几年的历史，十多岁的人正处在“青年时期”，远未定型．作为一门年轻的学

科，密码学正处在非常活跃的发展阶段，要覆盖住它的各个方面并非本书的目的．挂一漏万在所难免．例如目前讨论得如火如荼的零知识问题，本书仅一带而过，更多的问题只好割爱了．如果能做到让读者了解到密码学这个学科的梗概，而不感到不耐烦，作者就很满意了．

编 后 记

1989年夏，国内一些数学家和湖南教育出版社编辑同志在南开大学和北京大学聚会，深深感到“当今数学的面貌日新月异，数学的功能正在向其他自然科学、工程技术甚至社会科学领域扩展和渗透，数学本身在强大的社会要求和内部动力的推动下，不断追求自身的发展和完美”，希望能组织各方面专家编写一批书籍.“在中学数学的基础上，用现代观点向高中生、中学教师、大学生、工程技术人员、自然科学和社会科学工作者以及一切数学爱好者介绍一些数学思想，使大家真正地认识数学，了解数学，热爱数学，走向数学”. 这就是“走向数学”丛书的起源. 我们商定这套通俗读物的宗旨是：“用浅显易懂的语言从各个方面和角度向读者展示一些重要的数学思想，讲述数学（尤其是现代数学）的重要发展，介绍数学新兴领域、数学的广泛应用以及数学史上主要数学家（包括我国数学家）的成就.”

由于数学界大力支持、“数学天元项目”的赞助和各方面热情协助，一年后，第一辑八本书已与读者见面，第二辑也即将出版. 这十六本书尽管深浅不同，风格各异，但至少有一个共同之处，即作者们均朝着本丛书的宗旨和目标作了认真的努力.

在这批书中，作者们介绍了近年来数学一些重要发展和新的方向（其中包括1990年费尔兹奖获得者 V. Jones 在拓扑学纽

结理论方面的杰出工作，拓扑学家 Kuhn 和 Smale 在数值复杂性方面的开创性工作，实动力系统的奠基性结果等），以中学数学为起点介绍一些数学分支和课题（如复函数、非欧几何，有限域、凸性、拉姆塞理论、Polya 计数技术等），通过具体实例引伸出重要的数学思想和方法(如数论在数值计算中的应用，几何学的近代观点，群在集合上的作用，计算的复杂性概念等)，从不同的侧面介绍了数学在物理、化学、经济学、信息科学以及工农业生产等方面的广泛应用，包括华罗庚教授多年来在中国普及数学方法的宝贵经验．在书的正文或附录中，作者们介绍了中外许多数学家的生平和业绩，特别是国内外数学家为华罗庚教授所写的纪念文章，从不同侧面回忆了他早年的业绩，赞扬他为新中国培养人材和热爱祖国献身事业的可贵精神，这对于我们(包括年轻一代)是有很大教育意义的.

尽管作者们作了很大的努力，但我们深知，用通俗语言介绍如此丰富的数学思想和飞跃的发展，是一项十分艰难的任务，在第一批书出版之后，我们热诚地欢迎广大读者的批评和意见，以利于今后的改进和提高，如前所述，这批书的写作风格各异，取材的深度和广度也有所差别，即使不少作者几易其稿，力图把基点放在初等数学，但是要介绍现代数学的思想和内容，很难避免引进深一层的概念和方法，所以，我们不能苛求读者在最初几遍就能把书中叙述的内容和体现的思想方法全部读懂，但是希望具有不同程度数学知识和修养的数学爱好者在认真读过这些书之后都能有所收获，开阔眼界，增长见识，从而更加认识数学，了解数学，热爱数学和走向数学.

冯克勤

识于一九九二年五月